



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

**Clasificación de reclamos virtuales en el sector público:
un análisis comparativo de algoritmos de aprendizaje
automático usando procesamiento de lenguaje natural**

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Gestión de Tecnología de
Información y Comunicaciones

AUTOR

Thalia DIAZ TUNJAR

ASESOR

Dr. José Alfredo HERRERA QUISPE

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Díaz, T. (2025). *Clasificación de reclamos virtuales en el sector público: un análisis comparativo de algoritmos de aprendizaje automático usando procesamiento de lenguaje natural*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Thalia Diaz Tunjar
Tipo de documento de identidad	DNI
Número de documento de identidad	70813325
URL de ORCID	https://orcid.org/0000-0002-2402-7940
Datos de asesor	
Nombres y apellidos	José Alfredo Herrera Quispe
Tipo de documento de identidad	DNI
Número de documento de identidad	40362859
URL de ORCID	https://orcid.org/0000-0002-8207-9714
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	David Santos Mauricio Sánchez
Tipo de documento	DNI
Número de documento de identidad	06445495
Miembro del jurado 1	
Nombres y apellidos	Jorge Leoncio Guerra Guerra
Tipo de documento	DNI
Número de documento de identidad	08473333
Miembro del jurado 2	
Nombres y apellidos	Hugo David Calderon Vilca
Tipo de documento	DNI
Número de documento de identidad	01317748
Datos de investigación	

Línea de investigación	C.0.5.8
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Lima Latitud: -12.055926 Longitud: -77.083469
Año o rango de años en que se realizó la investigación	2022 - 2025
URL de disciplinas OCDE	Informática y Ciencias de la Información https://purl.org/pe-repo/ocde/ford#1.02.00 Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01 Ciencias de la información https://purl.org/pe-repo/ocde/ford#1.02.02



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER
EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON MENCIÓN EN GESTIÓN DE
TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIONES**

A los trece (13) días del mes de octubre de 2025, siendo las 12:00 p.m., se reunieron en el Auditorio Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. David Santos Mauricio Sánchez (Presidente)
Mg. Jorge Leoncio Guerra Guerra (Miembro)
Dr. Hugo David Calderon Vilca (Miembro)
Dr. José Alfredo Herrera Quispe (Miembro Asesor)

Se inició la Sustentación invitando a la candidata a Magister **THALIA DIAZ TUNJAR**, para que realice la exposición oral de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones, siendo la Tesis intitulada:

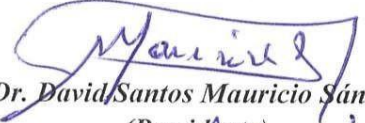
“Clasificación de reclamos virtuales en el sector público: un análisis comparativo de algoritmos de aprendizaje automático usando procesamiento de lenguaje natural”.

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

..... *avince* (15) *BUGNO*

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Bachiller **THALIA DIAZ TUNJAR** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

Siendo las *13:51* horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.


Dr. David Santos Mauricio Sánchez
(Presidente)


Dr. Hugo David Calderon Vilca
(Miembro)


Mg. Jorge Leoncio Guerra Guerra
(Miembro)


Dr. José Alfredo Herrera Quispe
(Miembro Asesor)

CERTIFICADO DE SIMILITUD

Yo, José Alfredo Herrera Quispe en mi condición de asesor acreditado con Dictamen N° 000004-2022-UPG-VDIP-FISI/UNMSM de la tesis, cuyo título es “Clasificación de reclamos virtuales en el sector público: un análisis comparativo de algoritmos de aprendizaje automático usando procesamiento de lenguaje natural”, presentado por el bachiller Thalia Diaz Tunjar, para optar el grado académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones,

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 2 % de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

Lima, 07 de febrero de 2025



Firma del Asesor

Nombres y Apellidos del asesor: José Alfredo Herrera Quispe
DNI: 40362859

DEDICATORIA

A mis padres, por haberme enseñado los principios y valores que llevo conmigo y por su incondicional apoyo.

A mis hermanos, por su constante respaldo en mi desarrollo profesional.

AGRADECIMIENTOS

Al Dr. José Alfredo Herrera Quispe, mi asesor, quiero agradecerle por su constante apoyo, orientación y revisión en la elaboración de este trabajo de investigación.

A mis padres, les agradezco su apoyo constante en todo el proceso de mi educación y por ser mi mayor soporte en la vida.

A mis hermanos, les agradezco su apoyo incondicional para alcanzar mis objetivos personales y profesionales.

A los profesores de mi maestría en la Universidad Nacional Mayor de San Marcos, les agradezco por compartir sus conocimientos durante los dos años de estudio.

Índice General

RESUMEN	VIII
ABSTRACT	IX
I Introducción	1
1.1 Situación problemática	1
1.2 Formulación del problema	4
1.2.1 Problema general	4
1.2.2 Problemas específicos	4
1.3 Justificación Teórica de la Investigación	5
1.4 Justificación Práctica de la Investigación	5
1.5 Objetivos de la investigación	6
1.5.1 Objetivo general	6
1.5.2 Objetivos específicos	7
1.6 Hipótesis de la investigación	7
1.6.1 Hipótesis general	7
1.6.2 Hipótesis específicas	7
II Marco Teórico	8
2.1 Revisión Sistemática de la Literatura	8
2.1.1 Clasificación de Quartil	10
2.2 Antecedentes de investigación	10
2.3 Bases Teóricas	15
2.3.1 Machine Learning (aprendizaje automático)	15
2.3.2 Perspectivas en el Machine Learning	16
2.3.3 Claves del aprendizaje automático (ML)	16
2.3.4 Algoritmos de Aprendizaje Automático (ML)	17
2.3.5 Procesamiento de Lenguaje Natural	19
2.4 Marco conceptual	20
2.4.1 Libro de reclamaciones virtual	20
2.4.2 Vectores de soporte y de aprendizaje automático (ML) para documentos	21

2.4.3	Clasificación de texto mediante un enfoque supervisado de aprendizaje automático (ML)	22
2.4.4	Clasificación Supervisada y Entrenamiento	24
2.4.5	Árboles de Decisión	25
2.4.6	Random Forests	27
2.4.7	Regresión Logística	29
2.5	Glosario	30
III	Metodología	32
3.1	Tipo y Diseño de la Investigación	32
3.1.1	Nivel de Investigación	32
3.1.2	Tipo de Investigación	32
3.1.3	Diseño de la Investigación	32
3.1.4	Método de la investigación.	32
3.2	Dataset y Partición	33
3.2.1	Dataset	33
3.2.2	Partición del Dataset	33
3.2.3	Técnica de muestreo de partición	33
3.3	Operacionalización de Variables	34
3.3.1	Identificación de Variables	34
3.3.2	Matriz de Operacionalización de Variables	34
3.3.3	Descripción de Indicadores de Variables	38
3.4	Justificación de los Algoritmos de Clasificación	38
3.5	Configuración e inicialización de los algoritmos	40
3.5.1	Naive Bayes (NB)	40
3.5.2	Linear Support Vector Classifier (LinearSVC)	40
3.5.3	Random Forest (RF)	41
3.5.4	Regresión Logística (LR)	41
3.5.5	Modelos de Lenguaje Grande (LLM) – Claude AI	41
3.6	Procedimientos	43
3.6.1	Arquitectura General del Proceso	43
3.6.2	Experimentos de Validación para las Hipótesis	43
3.6.3	Validación para las Hipótesis Específicas	45
IV	Fase Experimental	47
4.1	Observación previa del dataset <i>Reclamos_DT</i> .	47
4.2	Privacidad y confidencialidad de los datos	48
4.3	Preprocesamiento de NPL	48
4.3.1	Inspección Inicial	49
4.3.2	Limpieza del dataset	49

4.3.3	Estandarización de Etiquetas	49
4.3.4	Tokenización y Vectorización TF-IDF	51
4.3.5	Vectorización Sentence Embedding con Transformers	52
4.3.6	Partición	53
4.4	Entrenamiento de los Modelos	54
4.5	Balanceo de Datos	55
4.6	Experimentación para las Hipótesis	55
4.6.1	Naive Bayes (alpha = 0.0098, fit_prior = True)	55
4.6.2	LinearSVC (C = 1.0, loss = "squared_hinge", dual = False, max_iter = 2000, random_state = 42)	56
4.6.3	Random Forest (n_estimators = 100, min_samples_leaf = 10, max_features = 'sqrt', random_state = 42)	57
4.6.4	Regresión Logística (solver = "lbfgs", penalty = "l2", C = 7, max_iter = 1000, random_state = 42)	57
4.6.5	Agente de Claude AI	58
4.7	Simulación de Modelo de Clasificación Manual	58
V	Resultados y Discusión	60
5.1	Resultados de la Clasificación	60
5.1.1	Comparación Cruzada por Categoría	60
5.1.2	Mejor Modelo	61
5.1.3	Tiempos de entrenamiento y consideración computacional	62
5.2	Impacto en la Gestión Pública	63
5.3	Discusión de Resultados	63
5.4	Validación de Hipótesis	64
VI	Conclusiones y Recomendaciones	66
6.1	Conclusiones	66
6.1.1	Conclusión General	66
6.1.2	Conclusiones Específicas	66
6.2	Recomendaciones	68
Anexos		73

Lista de Tablas

Tabla 1.	Distribución de referencias bibliográficas por cuartiles según año de publicación	10
Tabla 2.	Matriz de operacionalización de variables.	35
Tabla 3.	Distribución completa de la variable <i>Tipo de registro</i>	47
Tabla 4.	Estructura inicial del conjunto de datos	49
Tabla 5.	Verificación de valores nulos en el conjunto de datos original	49
Tabla 6.	Distribución de la variable <i>Registro</i> tras la estandarización de etiquetas	50
Tabla 7.	Registros clasificados por su categoría	50
Tabla 8.	Fragmento ilustrativo de la matriz de embeddings	53
Tabla 9.	Distribución de los conjuntos de datos por método de vectorización	54
Tabla 10.	Reporte de clasificación en validación - Naive Bayes	55
Tabla 11.	Reporte de clasificación en validación - LinearSVC	56
Tabla 12.	Reporte de clasificación en validación - Random Forest	57
Tabla 13.	Reporte de clasificación en validación - Logistic Regression	57
Tabla 14.	Reporte de clasificación en validación - Claude AI	58
Tabla 15.	Estadísticas del tiempo de clasificación manual (N = 50,000 casos)	59
Tabla 16.	Resultados por tipo de mensaje - Tiempo de clasificación	59
Tabla 17.	Comparación de los modelos en la categoría <i>Reclamo</i>	60
Tabla 18.	Comparación de los modelos en la categoría <i>Sugerencia</i>	61
Tabla 19.	Desempeño del modelo <i>Regresión Logística</i> para clasificar categorías	61
Tabla 20.	Tiempos de entrenamiento observados (experimentales)	62
Tabla 21.	Recursos experimentales: <i>dataset</i> y código fuente para su uso libre	74

Lista de Figuras

Figura 1	Plazo de atención de los reclamos virtuales en las entidades públicas del Perú	1
Figura 2	Tiempo promedio de atención de los reclamos virtuales en las entidades públicas del Perú	2
Figura 3	Percepción del ciudadano sobre la atención brindada por medio del libro virtual de reclamos	3
Figura 4	Los vectores de soporte son los 5 puntos justo frente al margen del clasificador.	21
Figura 5	Conceptualización de la clasificación de textos mediante un enfoque supervisado de aprendizaje automático (ML).	22
Figura 6	Clasificación supervisada	24
Figura 7	Ejemplo de un árbol de decisión binario.	25
Figura 8	Ejemplo de dos árboles de decisión aleatorizados.	26
Figura 9	Ejemplo de árbol de decisión aleatorizado.	27
Figura 10	Ejemplo de Fronteras de decisión para un conjunto de árboles de decisión aleatorios (Random Forests).	28
Figura 11	Ejemplo de Datos para regresión con Random Forests.	29
Figura 12	Arquitectura general del modelo propuesto.	43
Figura 13	Preprocesamiento del Dataset.	48
Figura 14	Entrenamiento de los Modelos.	55

RESUMEN

La atención oportuna y eficiente de los reclamos ciudadanos en el sector público representa un desafío persistente, especialmente ante el creciente volumen de registros en plataformas virtuales. El procesamiento manual de estos reclamos genera demoras, errores y respuestas insatisfactorias que afectan la confianza del ciudadano en la gestión pública. La presente investigación de carácter aplicado y experimental tiene como objetivo validar la efectividad de técnicas de clasificación automática de reclamos mediante el uso de *Natural Language Processing (NLP)* y algoritmos de *Machine Learning (ML)*. Para ello, se formula una hipótesis principal orientada a determinar si es posible mejorar la clasificación y gestión de reclamos en un libro de reclamaciones virtual mediante la aplicación de modelos supervisados entrenados con datos reales del sector público peruano. Se desarrolló un análisis comparativo de cinco algoritmos de clasificación: *Naive Bayes*, *Random Forest*, *Logistic Regression*, *Linear Support Vector Classifier (LinearSVC)* y el modelo de lenguaje *Claude AI*. El conjunto de datos constó de 261,627 registros textuales, categorizados en seis clases: reclamo, queja, denuncia, consulta, sugerencia y otro. Los textos fueron sometidos a procesos de lematización, vectorización *TF-IDF* y balanceo mediante *SMOTE*, y los modelos fueron evaluados mediante métricas como precisión, *recall* y *F1-score*. Los resultados demuestran que es factible mejorar significativamente la clasificación. Entre los algoritmos clásicos, la *Logistic Regression* mostró un desempeño sólido y equilibrado, alcanzando un *F1-score* de 0.8639 en la categoría “Reclamo”. El modelo *LinearSVC* demostró alta capacidad de generalización con *recall* de 1.0 en clases minoritarias. El modelo *Claude AI* presentó un rendimiento competitivo, aunque inferior a los modelos supervisados optimizados. Esto confirma que la aplicación de *NLP* y *ML*, particularmente con modelos entrenados específicamente para el dominio, constituye una solución escalable y confiable para optimizar la gestión de reclamos, reduciendo tiempos de atención y aumentando la eficiencia operativa en entornos institucionales digitales.

Palabras clave: Clasificación automatizada, Procesamiento de lenguaje natural (NLP), Aprendizaje automático (ML), Libro de reclamaciones virtual, Sector público, Algoritmos de clasificación

ABSTRACT

The timely and efficient handling of citizen complaints in the public sector represents a persistent challenge, especially given the growing volume of records on virtual platforms. Manual processing of these complaints leads to delays, errors, and unsatisfactory responses that undermine citizen trust in public management. This applied and experimental research aims to validate the effectiveness of automatic complaint classification techniques using *Natural Language Processing (NLP)* and *Machine Learning (ML)* algorithms. To this end, a main hypothesis is formulated to determine whether it is possible to improve the classification and management of complaints in a virtual complaint book through the application of supervised models trained with real data from the Peruvian public sector. A comparative analysis of five classification algorithms was developed: *Naive Bayes*, *Random Forest*, *Logistic Regression*, *Linear Support Vector Classifier (LinearSVC)*, and the language model *Claude AI*. The dataset consisted of 261,627 textual records, categorized into six classes: claim, complaint, report, inquiry, suggestion, and other. The texts underwent lemmatization, *TF-IDF* vectorization, and balancing using *SMOTE*, and the models were evaluated using metrics such as precision, *recall*, and *F1-score*. The results demonstrate that significant improvement in classification is feasible. Among the classical algorithms, *Logistic Regression* showed solid and balanced performance, achieving an *F1-score* of 0.8639 in the Claim category. The *LinearSVC* model demonstrated high generalization capability with a *recall* of 1.0 in minority classes. The *Claude AI* model showed competitive performance, although inferior to the optimized supervised models. This confirms that the application of *NLP* and *ML*, particularly with models specifically trained for the domain, constitutes a scalable and reliable solution for optimizing complaint management, reducing processing times, and increasing operational efficiency in digital institutional environments.

Keywords: Automated classification, Natural language processing (NLP), Machine learning (ML), Virtual complaints book, Public sector, Classification algorithms

Capítulo I

Introducción

1.1. Situación problemática

El abundante registro de reclamos en las entidades se vuelve algo complejo de manejar a tiempo, teniendo en cuenta que el responsable de la entidad es el que debe abrir uno a uno estos reclamos, y según criterio, ver si corresponde ser atendido como reclamo. Esta es una tarea que se vuelve más tediosa cuando el plazo para al menos haber revisado ese reclamo luego es de tres días hábiles, y en muchas ocasiones no se ha ejecutado ninguna acción sobre el, por lo que se necesita de alguna manera ahorrar tiempo en detectar el tipo de reclamo, ya que de acuerdo a ello se tendrá diversos flujos. Un reclamo debe ser atendido en un máximo de 30 días hábiles luego del registro.

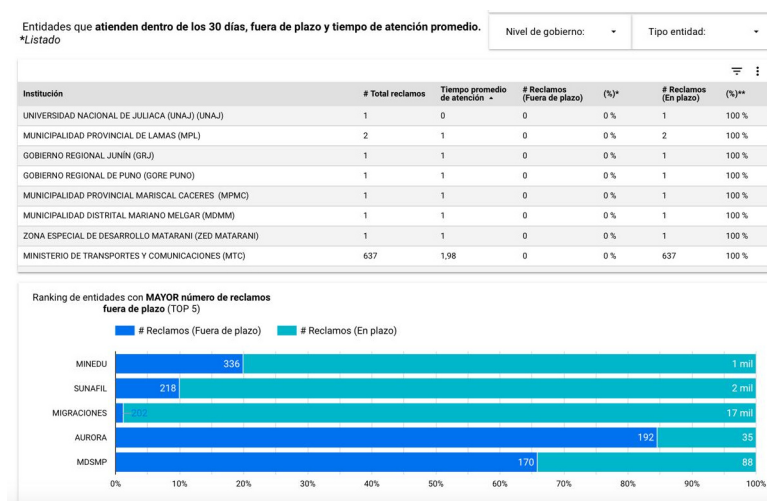


Figura 1

Plazo de atención de los reclamos virtuales en las entidades públicas del Perú
Fuente. Imagen tomada de Secretaría de Gestión Pública - PCM (2024).

En el gráfico podemos observar que hay entidades que pasan el tiempo de atención

de 30 días hábiles, haciendo un análisis del gráfico, por ejemplo, podemos notar que en el Programa Nacional para la Prevención y Erradicación de la Violencia contra las Mujeres e Integrantes del Grupo Familiar (AURORA) el número de reclamos de que fueron atendidos fuera de plazo asciende al 84,58 %, del total de reclamos recibidos de acuerdo a lo establecido en la Secretaría de Gestión Pública - PCM (2024)

También podemos notar que la Universidad Nacional Intercultural de la Selva Central Juan Santos Atahualpa figura entre las entidades con mayor tiempo promedio de atención mostrando su plazo máximo de 394 días hábiles.

Tiempo de atención

Presenta información del tiempo promedio de atención de los reclamos. Incluye ranking del menor y mayor tiempo promedio de atención.



Figura 2

Tiempo promedio de atención de los reclamos virtuales en las entidades públicas del Perú

Fuente. Imagen tomada de Secretaría de Gestión Pública - PCM (2024).

En cuanto a la percepción del ciudadano sobre la atención brindada y la respuesta recibida, es en mayor cantidad muy mala desde la creación de la plataforma hasta la actualidad; esto es uno de los factores que motivan esta investigación.



Figura 3

Percepción del ciudadano sobre la atención brindada por medio del libro virtual de reclamos

Fuente. Imagen tomada de Secretaría de Gestión Pública - PCM (2024).

La gestión eficiente de los reclamos de los ciudadanos es un desafío importante para muchas entidades públicas, especialmente en el contexto de un libro de reclamaciones virtual. En el pasado, estos reclamos se categorizaban y procesaban manualmente, lo cual era un proceso lento y propenso a errores, ineficiencias y sesgos. Sin embargo, la transición hacia métodos automatizados utilizando técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje automático (ML) promete mejorar la rapidez y precisión de estos procesos.

Además, la adopción de tecnologías emergentes como la inteligencia artificial (AI) para mejorar la participación ciudadana y la entrega de servicios públicos sugiere un cambio en cómo las organizaciones interactúan y responden a las necesidades de sus usuarios. En el contexto de un libro de reclamaciones virtual, la implementación de ML y NLP podría mejorar significativamente la comunicación con la administración pública y encontrar soluciones más efectivas a los problemas de los ciudadanos.

En el ámbito nacional, Ramirez-Guillen y Matzumura-Kasano (2024) analizaron 60 reportes de reclamos presentados durante el año 2022 en la consulta externa de un hospital público de Lima Norte. El estudio reveló que aproximadamente la mitad de los casos fue descartada por considerarse improcedente o sin sustento, mientras que la otra mitad derivó en distintas acciones por parte de la institución. Entre las respuestas implementadas destacaron las sanciones al personal involucrado, las compensaciones directas a los usuarios afectados y algunas medidas orientadas a fortalecer la calidad del servicio. A pesar de ello, los autores advierten que el manejo de estos reclamos responde principalmente a una lógica reactiva, sin estrategias claras de cómo gestionar

los reclamos para la mejora continua.

Si investigamos sobre experiencias internacionales exitosas en la gestión de reclamos, identificamos que muchos países siguen modelos que cuentan con las siguientes características: centradas en los ciudadanos, enfoque articulado entre entidades públicas, innovación y cultura y herramientas digitales de acuerdo con Autoridad Nacional del Servicio Civil (SERVIR) et al. (2022).

Por otro lado, Wang et al. (2020) investigaron la extracción de características y el análisis del procesamiento de lenguaje natural para el inglés, proponiendo un modelo híbrido que podría aplicarse a la segmentación de palabras, lo cual es relevante para la categorización automática de textos en reclamos. Este estudio subraya la relevancia de los procedimientos de extracción de características en ML y NLP para incrementar la eficacia y exactitud de los sistemas automáticos.

En este marco, el propósito de esta tesis es enfrentar el reto de categorizar automáticamente reclamaciones en un ambiente digital, a través de la implementación y adaptación de modelos de ML y NLP. El objetivo es investigar la efectividad de los métodos de Machine Learning en la comprensión semántica profunda y la segmentación de textos, con el objetivo de potenciar la capacidad de respuesta y la calidad del servicio al cliente. El estudio se enfocará en la creación de un sistema que, además de ser exacto y eficaz, también pueda ajustarse a las variaciones dinámicas y las demandas particulares del ámbito de los reclamos virtuales.

1.2. Formulación del problema

1.2.1. Problema general

¿Es factible mejorar la clasificación y gestión de reclamos en un sistema de libro de reclamaciones virtual mediante la aplicación de técnicas de aprendizaje automático (ML) y procesamiento de lenguaje natural (NLP)?

1.2.2. Problemas específicos

- ¿En qué medida puede un modelo de aprendizaje automático clasificar eficientemente los reclamos registrados en un libro de reclamaciones virtual, asegurando precisión y coherencia en las categorías asignadas?
- ¿Influirá la incorporación de técnicas de procesamiento de lenguaje natural en la comprensión y categorización automática de los textos asociados a reclamos ciudadanos?

1.3. Justificación Teórica de la Investigación

La presente investigación se fundamenta en desarrollos recientes en procesamiento de lenguaje natural (NLP) y aprendizaje automático (ML), dos disciplinas que han demostrado un impacto significativo en la automatización de tareas relacionadas con el análisis y categorización de textos en lenguaje natural. Su aplicación resulta especialmente pertinente en contextos institucionales donde se procesan grandes volúmenes de información ciudadana, como los sistemas de gestión de reclamos.

Numerosos estudios respaldan la efectividad del uso de técnicas de NLP y ML para abordar problemas complejos de clasificación textual. Por ejemplo, Olusegun et al. (2023) aplicaron modelos de categorización emocional para interpretar respuestas ciudadanas frente al brote de *monkeypox*, demostrando el potencial de estas herramientas para comprender demandas sociales expresadas en lenguaje natural. De forma complementaria, Özçift et al. (2021) evidenciaron la capacidad de modelos como BERT para tratar idiomas morfológicamente complejos sin necesidad de preprocesamiento intensivo, una característica de particular relevancia en contextos como el peruano, donde los textos de reclamos suelen contener expresiones informales o no estandarizadas.

Asimismo, Pereira et al. (2023) desarrollaron un sistema de búsqueda semántica basado en SBERT para mejorar la interacción entre ciudadanos y sistemas conversacionales, lo que refuerza la aplicabilidad de modelos de NLP en escenarios de atención pública. Por otro lado, Qurashi et al. (2023) abordaron la automatización de la gestión documental mediante aprendizaje automático en entornos organizacionales dinámicos, resaltando su utilidad en procesos que requieren actualización continua, como los reclamos virtuales.

Al situarse en la intersección entre la inteligencia artificial aplicada, la analítica institucional y la innovación en servicios públicos, esta investigación se justifica teóricamente en la necesidad de implementar soluciones automatizadas que optimicen los procesos administrativos del Estado. En ese marco, el uso de técnicas de NLP y ML representa no solo una tendencia tecnológica, sino una estrategia viable y sustentada para mejorar la interacción entre la ciudadanía y las instituciones públicas mediante la categorización automatizada de reclamos.

1.4. Justificación Práctica de la Investigación

El crecimiento exponencial de datos textuales en entornos digitales ha consolidado al aprendizaje automático (Machine Learning) y al procesamiento de lenguaje natural (Natural Language Processing, NLP) como herramientas clave para enfrentar desafíos

relacionados con la clasificación automatizada, el análisis semántico y la toma de decisiones basada en datos. En el contexto de la transformación digital del sector público, su aplicación en la gestión de reclamos ciudadanos resulta particularmente estratégica.

Muñoz-Céspedes et al. (2021) destaca la utilidad del NLP en la interpretación de percepciones y emociones expresadas por usuarios en plataformas digitales, aportando información valiosa para optimizar procesos de atención y respuesta institucional. Esta capacidad para analizar grandes volúmenes de datos no estructurados permite generar respuestas más informadas, oportunas y adaptadas a las necesidades reales del usuario.

Por su parte, Nirala et al. (2022) exploran el impacto de la inteligencia artificial conversacional en la atención ciudadana, destacando el uso de chatbots y sistemas automatizados que mejoran la eficiencia operativa, incrementan la precisión en la categorización de solicitudes y habilitan procesos de atención más personalizados. La generación de estos asistentes inteligentes depende críticamente de la disponibilidad de modelos entrenados en datos locales y específicos. Sin el desarrollo de investigaciones aplicadas como la presente —que construyen corpus clasificados y entrenan modelos ajustados al lenguaje y lógica institucional— es inviable avanzar hacia soluciones conversacionales contextualmente relevantes. Así, este trabajo sienta un precedente necesario para futuras aplicaciones conversacionales más sofisticadas en el ámbito público.

En este marco, la presente investigación propone el uso de modelos supervisados de clasificación combinados con técnicas de NLP para automatizar la categorización de reclamos ciudadanos en un entorno institucional. Esta solución tecnológica apunta a mejorar la eficiencia del proceso de registro y análisis de reclamos, reducir la intervención manual y aumentar la trazabilidad y coherencia de las respuestas institucionales. La propuesta se alinea con los desafíos actuales de modernización de la administración pública, ofreciendo una alternativa escalable, sostenible y basada en evidencia para fortalecer la interacción entre la ciudadanía y el Estado.

1.5. Objetivos de la investigación

1.5.1. Objetivo general

Mejorar la clasificación y gestión de reclamos en un sistema de libro de reclamaciones virtual, mediante el uso de técnicas de machine learning. clasificación automática de reclamos en un libro de reclamaciones virtual del sector público.

1.5.2. Objetivos específicos

- Diseñar y entrenar un modelo de machine learning para la clasificación eficiente de reclamos en un libro de reclamaciones virtual, asegurando precisión y relevancia en las categorías utilizadas.
- Utilizar técnicas de procesamiento de lenguaje natural (NLP) en de machine learning para mejorar la comprensión y categorización automática de los textos de reclamos.

1.6. Hipótesis de la investigación

1.6.1. Hipótesis general

La implementación y optimización de técnicas de machine learning mejora la clasificación y la gestión de reclamos en una plataforma de libro de reclamaciones virtual.

1.6.2. Hipótesis específicas

- El diseño y entrenamiento de un modelo de machine learning mejora la clasificación eficiente de reclamos en un libro de reclamaciones virtual, asegurando precisión y relevancia en las categorías utilizadas.
- La implementación de técnicas de procesamiento de lenguaje natural (NLP) en un sistema de machine learning mejora la comprensión y categorización automática de los textos de reclamos en un entorno virtual.

Capítulo II

Marco Teórico

2.1. Revisión Sistemática de la Literatura

Para la presente investigación, se realizó una revisión sistemática de la literatura (RSL) con el objetivo de identificar y analizar los estudios más relevantes sobre el uso del Procesamiento del Lenguaje Natural (PLN) en la mejora de la eficiencia de los servicios digitales, siguiendo las directrices metodológicas de Kitchenham y Charters (2009).

Fuentes de búsqueda: Se utilizaron bases de datos académicas reconocidas tales como Web of Science, IEEE Xplore, Scopus, ScienceDirect, Google Scholar y Taylor & Francis. Las búsquedas se realizaron empleando operadores booleanos y palabras clave como “*natural language processing*”, “*machine learning*”, “*digital services*”, y sus sinónimos.

Criterios de inclusión y exclusión:

- Artículos publicados entre 2019 y 2023.
- Estudios relacionados directamente con el uso de PLN para servicios digitales.
- Se excluyeron revisiones sistemáticas previas, estudios sin metodología definida o sin aplicación práctica clara.

Proceso de selección: Se identificaron inicialmente 544,029 artículos. Luego de aplicar filtros de relevancia, duplicidad y calidad metodológica, se seleccionaron finalmente 52 estudios para el análisis detallado.

Preguntas de investigación (RQ) y objetivos:

- **RQ1:** ¿Cuántos estudios se han publicado en los últimos cinco años sobre el tema?
Objetivo: Identificar la cantidad y tendencia temporal de las publicaciones.
- **RQ2:** ¿Cuáles son los principales medios de publicación?
Objetivo: Analizar la prevalencia en Journals o Conferences.
- **RQ3:** ¿Qué nivel de cuartil tienen las revistas donde se publicaron?
Objetivo: Determinar la calidad de las fuentes.
- **RQ4:** ¿Cuáles son las palabras clave más frecuentes?
Objetivo: Identificar focos temáticos recurrentes.
- **RQ5:** ¿Qué países lideran la producción científica en el tema?
Objetivo: Analizar la distribución geográfica de las investigaciones.

Resultados: Se evidenció un incremento constante en publicaciones relacionadas al PLN aplicado a servicios digitales. Predominaron revistas Q1 (59.6 %), y las palabras clave más frecuentes fueron “*natural language processing*”, “*machine learning*” y “*deep learning*”. Estados Unidos, China, India y Reino Unido lideraron la producción científica.

Estos hallazgos respaldan la relevancia actual del PLN como herramienta fundamental en procesos de transformación digital, y refuerzan el marco conceptual del presente trabajo.

2.1.1. Clasificación de Cuartil

Tabla 1

Distribución de referencias bibliográficas por cuartiles según año de publicación

Cuartil	Período	Cantidad	Referencias
Q1	2019-2020	4	Sarkar, 2019 , Giannopoulou y Mitrou, 2020 , Kowalski et al., 2020 , Wang et al., 2020
Q2	2021-2022	12	Peixoto, 2021 , Lopes-Cardoso et al., 2021 , Özçift et al., 2021 , Muñoz-Céspedes et al., 2021 , Søgaaard, 2021 , PCM, 2021 , IBM, 2021 , Ain et al., 2022 , Garcia Amador et al., 2022 , Stock et al., 2022 , Nirala et al., 2022 , Bozyiğit et al., 2022 , Autoridad Nacional del Servicio Civil (SERVIR) et al., 2022 , Deza y Atamtürk, 2022
Q3	2023	7	Flores Poma, 2023 , Naik et al., 2023 , Gao et al., 2023 , Olusegun et al., 2023 , Pereira et al., 2023 , Qurashi et al., 2023 , De Jager, 2023
Q4	2024-2025	9	Saha et al., 2024 , Houssein et al., 2024 , Barbierato y Gatti, 2024 , Ramirez-Guillen y Matzumura-Kasano, 2024 , Secretaría de Gestión Pública - PCM, 2024 , Blandón Andrade et al., 2025 , Schwaar et al., 2025
Total	2019-2025	32	

2.2. Antecedentes de investigación

Bozyiğit et al. ([2022](#)) desarrollaron un sistema de clasificación automática de reclamos en el sector alimentario utilizando técnicas de procesamiento de lenguaje natural y algoritmos de aprendizaje automático. El estudio empleó un conjunto de 2,217 reclamos ciudadanos recolectados a través de múltiples canales oficiales, como centros de atención, formularios web y correos institucionales. Los textos fueron preprocesados y representados mediante los métodos TF-IDF y word2vec, y posteriormente clasificados utilizando algoritmos como regresión logística, Naive Bayes, k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Random Forest y XGBoost. Los resultados mostraron que, en combinación con TF-IDF y reducción de dimensionalidad mediante Chi-cuadrado (CH2), estos modelos lograron métricas de desempeño consistentes, con F-measure superiores al 80 % en la mayoría de los casos. El estudio destaca que la correcta ingeniería de características y la selección de atributos relevantes pueden

mejorar significativamente la capacidad de los modelos clásicos para categorizar reclamos temáticamente, constituyendo un enfoque viable para su adopción en instituciones públicas con grandes volúmenes de datos textuales.

Peixoto (2021) propusieron una estrategia de clasificación automática de reclamos de consumidores usando técnicas de Minería de Textos y Aprendizaje de Máquina para apoyar la gestión del servicio de atención al cliente. El estudio empleó algoritmos de distinta complejidad, incluyendo Naive-Bayes SVM, redes neuronales tipo Multilayer Perceptron con embeddings FastText, y el modelo de lenguaje DistilBERT. A partir de datos extraídos de la plataforma ReclameAQUI, se evaluó la capacidad de los modelos para predecir la resolutividad de los reclamos, observando mejoras sustanciales cuando se incluyó tanto la narrativa del cliente como la respuesta de la empresa. Los resultados demostraron que el uso de técnicas de procesamiento de lenguaje natural puede aportar valor en entornos de toma de decisiones operativas, ayudando a reducir el riesgo reputacional de las organizaciones. El estudio también enfatiza la importancia de seleccionar modelos adecuados según la estrategia de negocio, y subraya su potencial para ser integradas en flujos automatizados de atención al cliente.

Saha et al. (2024) realizaron un trabajo enfocado en la clasificación automatizada y el enrutamiento de reclamos públicos utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático. La metodología se basa en el diseño de un sistema capaz de recibir textos de reclamos desde distintos canales (correos, formularios web, chats), realizar un preprocesamiento lingüístico, extraer características relevantes mediante TF-IDF y BERT, y clasificarlos usando modelos como Regresión Logística, SVM, Random Forest y BERT. Posteriormente, el sistema enruta los reclamos clasificados hacia las áreas correspondientes. Los resultados mostraron mejoras significativas en la eficiencia operativa y en la satisfacción del usuario final, gracias a la reducción de la carga manual y la capacidad de clasificar reclamos multilingües con precisión. Este estudio demuestra la aplicabilidad de modelos de lenguaje modernos y técnicas supervisadas en el tratamiento automático de reclamos textuales, lo cual es especialmente relevante en el contexto del sector público digital.

Flores Poma (2023) realizaron una revisión sistemática sobre la clasificación automática de quejas mediante procesamiento de lenguaje natural, analizando 24 artículos publicados entre 2018 y 2023 en bases como Scopus y Web of Science. Identificaron que los algoritmos más frecuentes fueron Support Vector Machine (15 %), GRU (12 %), BERT (12 %), MLP (9 %) y LSTM (9 %). Se destaca la importancia del preprocesamiento textual —como la tokenización, lematización y eliminación de palabras vacías— para mejorar el desempeño de los modelos. Las métricas más utilizadas fueron exactitud, precisión, recall y F1-score. Entre los conjuntos de datos analizados fi-

guran uno con 555,958 registros y otro con 20,000 entradas recolectadas en seis meses. El estudio concluye que, a pesar de los avances, aún persisten retos como el desbalance de clases, la escasez de datos etiquetados y la falta de mecanismos para manejar clases nuevas, lo cual limita la implementación efectiva en entornos del sector público.

Blandón Andrade et al. (2025) realizaron una revisión sistemática sobre métodos computacionales para el procesamiento de quejas expresadas en lenguaje natural. La metodología se basó en las guías de Kitchenham y PRISMA, permitiendo identificar 27 estudios relevantes publicados desde 2018. Los métodos analizados se clasificaron en cuatro categorías: lingüísticos (18.5 %), estadísticos (3.7 %), de aprendizaje automático (51.9 %) y métodos híbridos (25.9 %). Dentro de los enfoques híbridos, se reportaron precisiones superiores al 90 % en tareas de clasificación multiclase de quejas, como en el caso de algoritmos Naive Bayes y KNN aplicados a portales ciudadanos. Los autores concluyen que los métodos híbridos —que combinan técnicas de procesamiento de lenguaje natural con modelos de aprendizaje automático— ofrecen un enfoque prometedor para reducir tiempos de respuesta y mejorar la eficiencia en el tratamiento automatizado de reclamos, especialmente en contextos del sector público donde el cumplimiento normativo y la eficiencia operativa son críticos.

Lopes-Cardoso et al. (2021) propusieron un sistema robusto de procesamiento y clasificación automática de reclamos ciudadanos redactados en portugués, en colaboración con la Autoridad de Seguridad Alimentaria y Económica de Portugal (ASAE). La investigación abordó tres tareas principales: la identificación de la actividad económica implicada, la predicción de la gravedad de las infracciones y la determinación de la competencia institucional para su atención. A partir de un corpus real de 150,669 reclamos recolectados entre 2008 y 2018 a través de formularios web y correos electrónicos, se diseñaron modelos de clasificación multiclase y multietiqueta, usando SVM, Random Forest y métodos basados en lenguaje como BERT. El mejor modelo para la clasificación de actividad económica logró una exactitud de 74.6 % y macro-F1 de 56.4 %, mientras que en la predicción de gravedad de infracciones se alcanzó una exactitud del 70.1 % y macro-F1 de 65.4 %. Para la tarea de competencia institucional, el clasificador SVM logró un macro-F1 de 77.5 %. El estudio destaca la necesidad de explicabilidad en sistemas automáticos usados en contextos gubernamentales y demuestra que enfoques tradicionales bien afinados pueden competir con modelos complejos, siendo además más adecuados para su implementación operativa en el sector público.

Naik et al. (2023) desarrollaron un sistema automatizado para la clasificación de quejas utilizando técnicas de procesamiento de lenguaje natural, aprendizaje automático y aprendizaje profundo. El estudio utilizó un conjunto de datos extraído del sitio oficial del Consumer Financial Protection Bureau (CFPB), compuesto por reclamos categori-

zados en nueve clases. Tras aplicar preprocesamiento textual, lematización y técnicas de aumento de datos, evaluaron varios algoritmos de clasificación. Entre los modelos de aprendizaje automático, el mejor rendimiento fue alcanzado por SVM lineal y regresión logística, ambos con una exactitud de 85.35 % al usar representaciones bigram. En cuanto a aprendizaje profundo, una red CNN a nivel de palabra obtuvo una exactitud de prueba del 82.46 %, superando al modelo LSTM que alcanzó 77.86 %. Los resultados sugieren que, en contextos donde se busca equilibrio entre precisión y eficiencia computacional, los modelos basados en SVM y CNN ofrecen soluciones viables para la clasificación automática de quejas, facilitando la derivación de estas hacia el área correspondiente sin intervención humana.

Garcia Amador et al. (2022) desarrollaron un modelo de clasificación automática de peticiones, quejas, reclamos y sugerencias utilizando técnicas de procesamiento de lenguaje natural y algoritmos de aprendizaje automático. El estudio se basó en la recolección y análisis de más de 20,000 registros textuales provenientes de plataformas digitales de interacción con usuarios, los cuales fueron depurados y etiquetados manualmente en dos categorías principales: valoraciones positivas (32.5 %) y negativas (67.5 %). Se experimentó con múltiples modelos, entre ellos Naive Bayes Multinomial, SVM y redes neuronales sin embeddings preentrenados, alcanzando en validación sensibilidades superiores al 70 % y valores de AUC cercanos al 75 %. Los autores concluyen que, para contextos institucionales con altos volúmenes de retroalimentación ciudadana, los modelos ligeros como Naive Bayes pueden ofrecer un equilibrio óptimo entre precisión, costo computacional y facilidad de implementación.

Ain et al. (2022) propusieron un modelo automatizado para la clasificación de reclamos en instituciones públicas mediante técnicas de procesamiento de lenguaje natural y algoritmos de aprendizaje automático. El estudio utilizó dos conjuntos de datos textuales: uno local, conformado por 10,000 reclamos dirigidos a una entidad gubernamental de Bureau, con 124,472 registros depurados. Tras un riguroso preprocesamiento, se aplicaron técnicas de vectorización (Count Vectorizer y TF-IDF) y se evaluaron cinco clasificadores: Naive Bayes, KNN, Random Forest, SVM y regresión logística. Los mejores resultados se obtuvieron con SVM y regresión logística, alcanzando una exactitud del 86 % en el segundo conjunto de datos y hasta un 80 % en el conjunto local. El estudio concluye que este enfoque permite reducir significativamente la carga operativa y los errores humanos, facilitando la clasificación eficiente y precisa de grandes volúmenes de reclamos ciudadanos en instituciones públicas.

Houssein et al. (2024) realizaron una revisión exhaustiva sobre el uso de técnicas de procesamiento de lenguaje natural y aprendizaje automático para la extracción de información en textos clínicos no estructurados, particularmente en registros electrónicos

de salud (EHR). Aunque su enfoque está en el dominio biomédico, los desafíos abordados y las soluciones propuestas son altamente relevantes para entornos institucionales donde se procesan grandes volúmenes de texto narrativo, como los reclamos ciudadanos. El estudio analiza más de 100 investigaciones recientes, detallando la aplicación de algoritmos como SVM, Random Forest, redes neuronales recurrentes (RNN), LSTM y modelos de atención, así como herramientas de vectorización como TF-IDF y embeddings contextuales. Se destacan tareas clave como la extracción de entidades, desambiguación semántica, detección de relaciones y clasificación automática de documentos. Los autores concluyen que los modelos basados en aprendizaje automático superan a los métodos tradicionales en tareas de clasificación complejas y enfatizan la necesidad de marcos explicables y éticamente responsables, especialmente al tratar datos sensibles. Estos hallazgos resultan aplicables a la clasificación automatizada de reclamos textuales en el sector público, donde también se requiere precisión, transparencia y cumplimiento normativo.

Gao et al. (2023) propusieron un modelo basado en procesamiento de lenguaje natural para la detección de anomalías sistemáticas en narrativas de reclamos de consumidores. El enfoque se basa en dos etapas: primero, la clasificación de reclamos como “meritorios” o “no meritorios” mediante algoritmos como SVM, regresión logística, random forest, gradient boosting y perceptrón multicapa; y segundo, la identificación de patrones anómalos dentro de los reclamos clasificados como meritorios. Para el entrenamiento se utilizaron 2,849 registros. El mejor desempeño se obtuvo con el clasificador SVM bajo la featurización TF-IDF, alcanzando un F1-score promedio de 78.24 % en pruebas repetidas. Además, se evaluó una variante del método TF-IDF combinada con intensidades de sentimiento VADER, la cual, si bien mostró menor exactitud (hasta 63.96 %), permitió controlar la proporción de reclamos clasificados erróneamente como meritorios. El estudio introduce índices cuantitativos para la detección de anomalías, lo que permite identificar desviaciones sistemáticas en los datos y fortalecer la toma de decisiones en entornos regulatorios y de atención ciudadana.

Stock et al. (2022) realizaron un estudio teórico sobre los fundamentos y desafíos del uso de algoritmos de aprendizaje automático en contextos donde se manejan datos sensibles, haciendo énfasis en los riesgos para la privacidad en aplicaciones prácticas. Si bien el enfoque principal se centra en el ámbito médico, los principios abordados resultan altamente relevantes para sistemas de clasificación automatizada de reclamos ciudadanos en el sector público. El estudio describe diversos algoritmos supervisados y no supervisados, como SVM, Random Forest, redes neuronales profundas y clustering con k-means, destacando sus ventajas y limitaciones. Además, introduce conceptos clave como el aprendizaje explicable y la necesidad de asegurar la equidad y transparencia de los modelos. Los autores concluyen que para garantizar la legitimidad de

decisiones automatizadas, como la clasificación de reclamos administrativos, es esencial aplicar técnicas de explicabilidad (XAI) y asegurar el cumplimiento normativo en protección de datos personales, como exige el artículo 22 del Reglamento General de Protección de Datos (GDPR).

Schwaar et al. (2025) realizaron un estudio orientado a la clasificación automatizada de documentos administrativos en la administración pública alemana, utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático. El objetivo fue asignar de forma automática mensajes entrantes a 31 departamentos gubernamentales, a partir de un corpus de 1,169 documentos normalizados y 11,386 segmentos de texto. La investigación comparó cuatro variantes de pipelines de clasificación, combinando métodos de vectorización (TF, TF-IDF), normalización (L1 y L2), reducción de dimensionalidad y algoritmos como SVM, regresión logística, redes neuronales y autoencoders supervisados (SVAE). Los mejores resultados se obtuvieron con regresión logística y redes neuronales en modelos normalizados por documento, alcanzando exactitudes de hasta 89.9 % y F1-scores superiores al 88.8 %. Los autores concluyen que el enfoque basado en normalización, en lugar de segmentación, mejora significativamente la precisión en tareas de clasificación multiclase en entornos administrativos, lo que lo convierte en una estrategia aplicable para la clasificación eficiente de reclamos ciudadanos en instituciones públicas digitalizadas.

En suma, estas investigaciones evidencian el efecto revolucionario del NLP y el aprendizaje automático en diferentes áreas, proponiendo una transición hacia la toma de decisiones fundamentada en datos en ciudades inteligentes, cuidado de la salud y planificación urbana.

2.3. Bases Teóricas

2.3.1. Machine Learning (aprendizaje automático)

El aprendizaje automático (Machine Learning, ML) es una subárea clave de la inteligencia artificial centrada en el desarrollo de modelos capaces de identificar patrones y tomar decisiones basadas en datos, sin necesidad de programación explícita para cada tarea de acuerdo a las ideas de Barbierato y Gatti (2024). Esta capacidad resulta especialmente valiosa en contextos donde los datos no estructurados, como los textos en lenguaje natural, requieren procesamiento automatizado para tareas de clasificación, predicción o análisis semántico.

En el marco de esta investigación, el ML se aplica como un enfoque supervisado para la categorización automática de reclamos ciudadanos. A través del entrenamiento

con ejemplos etiquetados, los algoritmos aprenden a asignar nuevas instancias a clases previamente definidas, optimizando sus parámetros para minimizar errores de clasificación. Este proceso permite transformar el manejo manual y subjetivo de los reclamos en una estrategia sistematizada, escalable y basada en evidencia.

La combinación de técnicas de ML con el procesamiento de lenguaje natural (NLP) permite, además, incorporar características lingüísticas relevantes en el modelado, lo que mejora la capacidad del sistema para entender variaciones textuales y contextuales presentes en los reclamos institucionales.

2.3.2. Perspectivas en el Machine Learning

El aprendizaje automático comprende distintos enfoques según la disponibilidad y naturaleza de los datos, siendo los más relevantes el aprendizaje supervisado, no supervisado y por refuerzo. En esta investigación, el enfoque adoptado es el *aprendizaje supervisado*, dado que se dispone de un conjunto de reclamos previamente etiquetados que permiten entrenar modelos capaces de predecir categorías en nuevos registros.

El aprendizaje supervisado permite construir clasificadores a partir de pares entrada-salida, donde las entradas son textos vectorizados y las salidas son las clases asociadas (como reclamo, queja, sugerencia, etc.). Este tipo de aprendizaje ha demostrado ser altamente eficaz en tareas de procesamiento de lenguaje natural (NLP) orientadas a la categorización de texto, como se evidencia en trabajos recientes de Bozyigit et al. (2022) y Ain et al. (2022).

Aunque el aprendizaje no supervisado y por refuerzo han sido aplicados en contextos específicos de descubrimiento de temas y mejora continua de políticas, respectivamente, su aplicación directa en la clasificación supervisada multiclase de reclamos es limitada en el contexto institucional peruano, donde la trazabilidad y coherencia de las categorías es esencial. Por ello, el uso de modelos supervisados ofrece un balance adecuado entre precisión, explicabilidad y viabilidad operativa en entornos públicos digitales.

2.3.3. Claves del aprendizaje automático (ML)

El desarrollo de modelos de aprendizaje automático efectivos requiere atender a varios componentes fundamentales, los cuales inciden directamente en el rendimiento del sistema y en la calidad de las predicciones. Según Sarkar (2019), el proceso inicia con la preparación de datos, que incluye la limpieza, normalización y transformación de los textos para su posterior análisis.

En el contexto del procesamiento de lenguaje natural (NLP), uno de los pasos clave es la extracción de características textuales relevantes, como términos más frecuentes o estructuras sintácticas representativas. Estas características son vectorizadas para alimentar algoritmos de clasificación. En esta investigación, se utilizaron técnicas como TF-IDF, que permiten convertir el contenido semántico de los reclamos en vectores numéricos que capturan la importancia relativa de las palabras dentro del corpus.

Una correcta partición de los datos en conjuntos de entrenamiento y validación es igual de esencial para garantizar un entrenamiento eficaz del modelo y una evaluación objetiva de su desempeño. Las métricas empleadas incluyen precisión, *recall*, *F1-score* y exactitud, estas permiten analizar la capacidad del modelo para clasificar correctamente reclamos en sus respectivas categorías (reclamo, queja, sugerencia, etc.).

El entrenamiento y validación iterativa de los modelos permite ajustar hiperparámetros para minimizar errores y mejorar la generalización, evitando fenómenos como el sobreajuste. En entornos institucionales, donde la coherencia y trazabilidad de las clasificaciones es crítica, estas claves del aprendizaje automático aseguran que el sistema sea tanto confiable como replicable.

2.3.4. Algoritmos de Aprendizaje Automático (ML)

El aprendizaje automático proporciona un conjunto diverso de algoritmos capaces de resolver tareas de clasificación supervisada en entornos de alta dimensionalidad, como es el caso de la categorización automática de texto. En esta investigación se consideraron cuatro algoritmos ampliamente utilizados en problemas de procesamiento de lenguaje natural: Naive Bayes, Regresión Logística, Random Forest y Linear Support Vector Classifier (LinearSVC).

Naive Bayes (MultinomialNB)

Este modelo probabilístico se fundamenta en el teorema de Bayes y parte del supuesto de independencia condicional entre características. En clasificación de texto, esto significa que la presencia de una palabra se considera independiente de las demás, lo cual simplifica el modelado de la distribución de términos. La versión multinomial es especialmente adecuada para datos representados como frecuencias ponderadas, como TF-IDF, y permite una implementación eficiente sobre espacios de alta dimensionalidad, siendo ampliamente utilizada en tareas de categorización textual de acuerdo a Mansoori et al. (2025).

Regresión Logística

La regresión logística es un modelo lineal que estima la probabilidad de pertenencia a una clase aplicando una función sigmoide sobre una combinación lineal de las características del texto. En procesamiento de lenguaje natural, este algoritmo se utiliza frecuentemente por su capacidad de modelar relaciones binarias y multiclase de manera eficiente. Su formulación probabilística facilita la interpretación de resultados, mientras que su compatibilidad con técnicas de vectorización como TF-IDF permite transformar entradas textuales en representaciones numéricas aptas para clasificación supervisada según Bhartendoo Vimal y RV College of Engineering (2020).

Random Forest

Random Forest es un método de ensamblado que combina múltiples árboles de decisión contruidos sobre subconjuntos aleatorios de datos y características. Su diseño reduce la varianza del modelo y mejora la capacidad de generalización, especialmente útil en tareas de clasificación de texto con alta dimensionalidad. Además, incorpora técnicas de selección de atributos relevantes, lo que permite mejorar el rendimiento sobre vectores generados mediante representaciones como TF-IDF de acuerdo a los lineamientos de Jalal et al. (2022).

Linear Support Vector Classifier (LinearSVC)

LinearSVC es un modelo lineal que busca un hiperplano óptimo para separar clases con el mayor margen posible en espacios de alta dimensionalidad. Su formulación es eficiente tanto en tiempo como en memoria, lo que lo hace especialmente adecuado para tareas de clasificación de texto. Además, puede extenderse a problemas multiclase mediante enfoques como “uno-contra-el-resto”, y su compatibilidad con técnicas como TF-IDF lo convierte en una opción robusta para la categorización automática de documentos según lo explica Sabiri et al. (2023).

Modelos de Lenguaje de Gran Escala (LLM)

Los modelos de lenguaje de gran escala representan un avance reciente en el procesamiento del lenguaje natural, basado en arquitecturas neuronales profundas -subgrupo del machine learning conocido como deep learning- que aprenden representaciones contextuales a partir de grandes volúmenes de texto, es decir, procesamiento de lenguaje natural a gran escala. Los LLM poseen una buena capacidad para comprender dependencias semánticas y generar predicciones coherentes en tareas complejas de clasificación y análisis textual. De acuerdo con Saha et al. (2024) y Blandón Andrade et al. (2025).

Cada uno de estos algoritmos representa una estrategia diferente para abordar la clasificación de texto, desde enfoques probabilísticos hasta métodos basados en márgenes o

ensamblados. Su estudio permite comprender las distintas formas de modelar relaciones complejas entre lenguaje natural y categorías institucionales, habilitando el diseño de sistemas de gestión automatizada basados en datos reales.

2.3.5. Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural (NLP, por sus siglas en inglés) es una subdisciplina de la inteligencia artificial que se centra en el análisis computacional del lenguaje humano. Su objetivo principal es dotar a los sistemas informáticos de la capacidad de interpretar, generar y razonar sobre datos expresados en lenguaje natural, tanto en forma escrita como oral (Søgaard, 2021). Entre las tareas más relevantes del NLP se encuentran la clasificación de texto, extracción de información, análisis de sentimientos, reconocimiento de entidades y traducción automática.

Una de las principales fortalezas del NLP es su capacidad para enfrentar la ambigüedad semántica, la variabilidad sintáctica y el ruido inherente al lenguaje humano. Como destaca De Jager (2023), el “ruido semántico” no debe entenderse como una anomalía, sino como un componente natural del lenguaje que los modelos deben aprender a gestionar, especialmente en entornos institucionales donde los textos pueden ser informales, incompletos o redactados sin estandarización.

En contextos aplicados, como los servicios públicos, el NLP ha mostrado un impacto significativo. Kowalski et al. (2020) demuestran su eficacia en la administración pública para analizar y clasificar textos provenientes de la ciudadanía. Estas aplicaciones confirman el potencial del NLP para transformar grandes volúmenes de información no estructurada en insumos valiosos para la toma de decisiones.

Desde una perspectiva técnica, el uso de modelos basados en redes neuronales profundas, como destaca Giannopoulou y Mitrou (2020), ha permitido mejorar significativamente la clasificación temática de documentos. Este avance es especialmente relevante en tareas como la clasificación de reclamos institucionales, donde los modelos deben captar matices léxicos y semánticos en descripciones breves y heterogéneas.

En el presente estudio, el NLP se emplea como herramienta clave para vectorizar y analizar descripciones textuales provenientes del libro de reclamaciones virtual, facilitando su clasificación automática en categorías predefinidas mediante algoritmos de aprendizaje supervisado.

2.4. Marco conceptual.

2.4.1. Libro de reclamaciones virtual

El libro de reclamaciones es la plataforma digital que soporta la gestión de reclamos, la cual permite a las personas registrar y hacer seguimiento a la atención de su reclamo, conociendo el estado en el que se encuentra; así como a las entidades y empresas públicas efectuar la gestión de reclamos, la supervisión de la atención, la obtención de información detallada y estadística para la toma de decisiones de la Alta Dirección, que contribuyan a la mejora continua de los bienes y servicios que prestan las entidades según PCM, [2021](#). El plazo de atención para cada reclamo registrado es de un máximo de treinta (30) días hábiles, computados desde el día siguiente de su recepción.

SMOTE (Synthetic Minority Oversampling Technique)

El *Synthetic Minority Oversampling Technique* (SMOTE) es un método de sobremuestreo propuesto para mitigar el desbalance de clases en problemas de aprendizaje supervisado. Su principio consiste en generar nuevas instancias sintéticas de la clase minoritaria mediante interpolación lineal entre un ejemplo x_0 y uno de sus K -vecinos más cercanos x_k , de acuerdo con la expresión:

$$z = (1 - w)x_0 + wx_k, \quad \text{con } w \sim U(0, 1),$$

donde z representa la nueva instancia generada y w es un valor aleatorio uniformemente distribuido que controla el grado de interpolación. Esta formulación permite generar un conjunto de patrones sintéticos que expanden el espacio de representación de la clase minoritaria sin replicar instancias existentes.

Desde un enfoque probabilístico, la distribución de las muestras generadas por SMOTE no necesariamente coincide con la densidad original de la clase minoritaria $p_X(x)$. En un estudio teórico reciente, Elreedy et al., [2024](#) derivan formalmente la función de densidad de probabilidad $p_Z(z)$ de las muestras sintéticas, demostrando que la técnica genera un desplazamiento de masa hacia el interior del espacio de la clase minoritaria, lo que implica una contracción de la distribución respecto a $p_X(x)$. Dicha función se define como:

$$p_Z(z) = (N - K) \binom{N - 1}{K} \int_x p_X(x) \int_{\|z - x\|}^{\infty} p_X \left(x + \frac{(z - x)r}{\|z - x\|} \right) \frac{r^{d-2}}{\|z - x\|^{d-1}} \\ \times B(1 - I_B(x, r); N - K - 1, K) dr dx$$

donde $B(\cdot)$ es la función beta incompleta e $I_B(x, r) = \int_{B(x, r)} p_X(x) dx$ representa la

cobertura de la clase minoritaria dentro de la esfera de radio r centrada en x . Este resultado constituye la primera formulación analítica exacta de la densidad inducida por SMOTE, permitiendo comprender su comportamiento desde la perspectiva de la teoría de distribuciones.

El valor de este análisis radica en demostrar que, aunque SMOTE mejora la representación de la clase minoritaria, su mecanismo de interpolación no preserva completamente la estructura estadística del conjunto original. Sin embargo, al aproximar razonablemente la densidad $p_X(x)$, el método logra reducir el sesgo hacia la clase mayoritaria y mejora la generalización de clasificadores en contextos de datos desbalanceados.

2.4.2. Vectores de soporte y de aprendizaje automático (ML) para documentos

La mejora de la efectividad de los clasificadores ha sido un área de investigación intensiva en el aprendizaje automático durante las últimas dos décadas. Este esfuerzo ha conducido al desarrollo de una nueva generación de clasificadores de vanguardia, incluyendo las máquinas de vectores de soporte (SVM), árboles de decisión potenciados, regresión logística regularizada, redes neuronales y Random Forests. Muchos de estos métodos, especialmente las SVM, han sido aplicados exitosamente a problemas de recuperación de información, particularmente en la clasificación de textos. Una SVM es un tipo de clasificador de gran margen que opera en un espacio vectorial con el objetivo de encontrar un límite de decisión entre dos clases que esté lo más lejos posible de cualquier punto en el conjunto de datos de entrenamiento, considerando algunos puntos como ruido o excepciones, de acuerdo a las ideas de Manning et al., [2008](#).

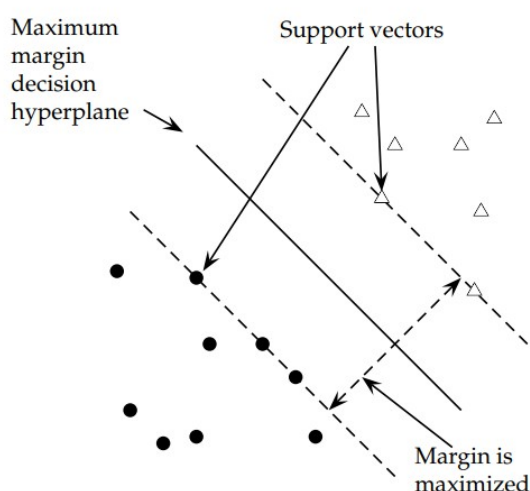


Figura 4

Los vectores de soporte son los 5 puntos justo frente al margen del clasificador.

Fuente. Tomado de Manning et al., [2008](#)

Asimismo, también indican que el desarrollo de las SVM se inicia con conjuntos de datos de dos clases que son separables por un clasificador lineal, para luego extender el modelo a datos no separables, problemas de múltiples clases y modelos no lineales, incluyendo una discusión adicional sobre el rendimiento de las SVM. Posteriormente, se considera la implementación práctica de clasificadores de texto: qué tipos de clasificadores son apropiados en diferentes situaciones y cómo se pueden explotar las características específicas del texto en la clasificación. Finalmente, se explora cómo la tecnología de aprendizaje automático desarrollada para la clasificación de texto puede aplicarse al problema de aprender a clasificar documentos en la recuperación ad hoc. Aunque se han aplicado varios métodos de aprendizaje automático a esta tarea, el uso de SVM ha sido prominente. Las máquinas de vectores de soporte no son necesariamente superiores a otros métodos de aprendizaje automático, excepto quizás en situaciones con pocos datos de entrenamiento, pero actúan a un nivel de vanguardia y poseen un gran atractivo teórico y empírico actual.

2.4.3. Clasificación de texto mediante un enfoque supervisado de aprendizaje automático (ML)

Se centra en el proceso y las metodologías utilizadas para asignar documentos textuales a una o más categorías predefinidas. Este proceso implica trabajar con una amplia gama de datos textuales que pueden variar desde frases y oraciones hasta documentos completos obtenidos de diversas fuentes, como corpora y blogs. La clasificación de texto, también conocida como categorización de texto, se basa en identificar y asignar estas unidades textuales a categorías específicas mediante un enfoque supervisado de aprendizaje automático (Sarkar, 2019).

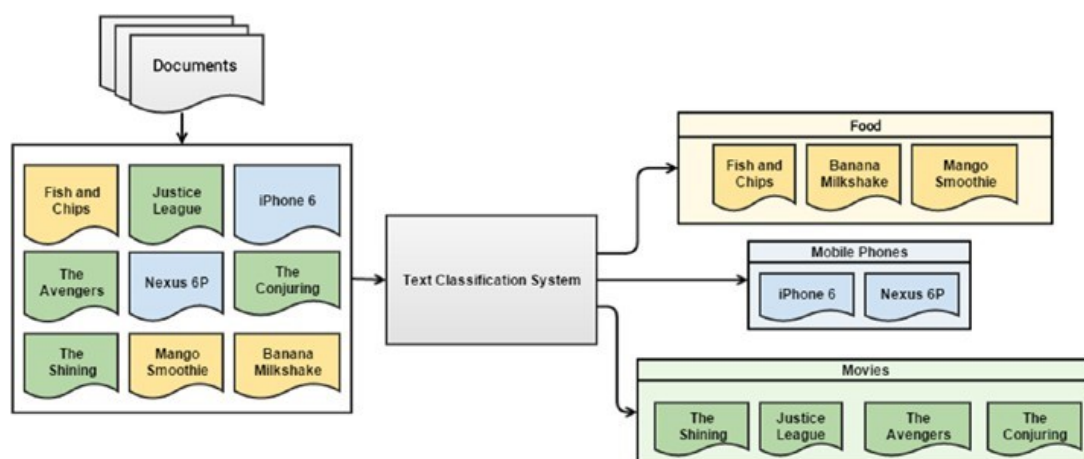


Figura 5

Conceptualización de la clasificación de textos mediante un enfoque supervisado de aprendizaje automático (ML).

Fuente. Imagen tomada de Sarkar (2019)

■ Definición y Alcance

- **Datos Textuales:** Incluyen todo tipo de frases, oraciones o los documentos completos con textos que pueden provenir de diferentes fuentes en la web.
- **Clasificación de Texto:** Proceso de asignar documentos a una o más categorías predefinidas. Se utiliza el término “documento” en un sentido amplio para abarcar cualquier representación concreta de pensamientos o eventos, ya sean escritos, grabaciones de voz, dibujos o presentaciones.

■ Metodología

- **Enfoque Supervisado:** Utiliza un conjunto de datos etiquetados para entrenar modelos de ML que puedan clasificar o categorizar nuevos documentos basándose en sus propiedades inherentes.

■ Tipos de Clasificación

- **Clasificación Basada en Contenido:** Prioriza o asigna pesos a temas específicos dentro del texto para determinar la categoría del documento.
- **Clasificación Basada en Solicitudes:** Se enfoca en las solicitudes de los usuarios, orientada hacia grupos específicos de usuarios y gobernada por políticas y principios específicos.

■ Proceso de Clasificación

El sistema de clasificación de texto asigna documentos a las categorías adecuadas basándose en sus atributos y propiedades, utilizando un espacio de alta dimensión para identificar y clasificar estos documentos eficazmente. Este proceso implica reconocer las características únicas de cada documento y utilizarlas para facilitar la identificación y clasificación correcta dentro de un sistema de clasificación de texto.

■ Aplicaciones

La clasificación de texto tiene aplicaciones en diferentes dominios, como la organización de productos en categorías específicas como alimentos, teléfonos móviles y películas, basándose en sus descripciones, atributos y otras propiedades que se utilizan como características en el sistema de clasificación.

2.4.4. Clasificación Supervisada y Entrenamiento

Bird et al. (2009) mencionan que la clasificación implica asignar la etiqueta de clase correcta a una entrada específica. En las tareas de clasificación estándar, cada entrada se evalúa de forma independiente, y el conjunto de etiquetas se establece previamente. Algunos ejemplos de tareas de clasificación incluyen:

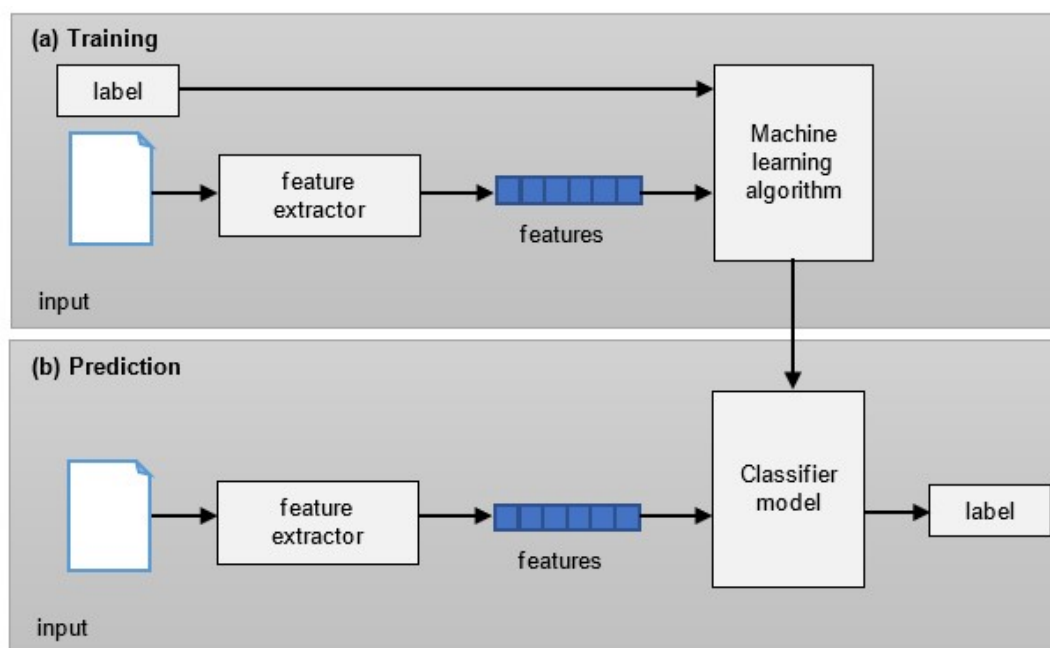
- Determinar si un correo electrónico es spam o legítimo.
- Identificar el tema de un artículo de noticias utilizando categorías predefinidas como “deportes”, “tecnología” y “política”.
- Identificar si una instancia de la palabra “banco” se refiere a la orilla de un río, una institución financiera, el acto de inclinarse hacia un lado, o la acción de depositar dinero en una institución financiera.

La tarea básica de clasificación presenta diversas variantes interesantes. Por ejemplo, en la clasificación multiclase, cada instancia puede tener múltiples etiquetas; en la clasificación de clase abierta, el conjunto de etiquetas no se define de antemano; y en la clasificación de secuencias, se clasifica conjuntamente una lista de entradas.

Un clasificador supervisado se desarrolla empleando un conjunto de datos de entrenamiento que incluye las etiquetas correctas para cada entrada. La estructura de clasificación supervisada muestra cómo se organiza este proceso.

Figura 6

Clasificación supervisada



Fuente. Tomado de Bird et al. (2009).

En la etapa de entrenamiento (a), se utiliza un extractor de características para convertir cada valor de entrada en un conjunto de características. Estos conjuntos contienen información clave sobre cada entrada que es esencial para su clasificación, y serán explicados en la sección siguiente. Los conjuntos de características junto con sus etiquetas correspondientes se entregan al algoritmo de aprendizaje automático, lo que permite construir un modelo. En la fase de predicción (b), se emplea el mismo extractor de características para procesar nuevas entradas y convertirlas en conjuntos de características. Estos conjuntos luego se introducen en el modelo, el cual genera las etiquetas predichas.

2.4.5. Árboles de Decisión

Los árboles de decisión representan una metodología altamente intuitiva para clasificar o etiquetar objetos mediante una serie de preguntas específicas que conducen a la clasificación deseada. Por ejemplo, al intentar clasificar un animal encontrado durante una caminata, podríamos desarrollar un árbol de decisión similar al ilustrado en la Figura 7.

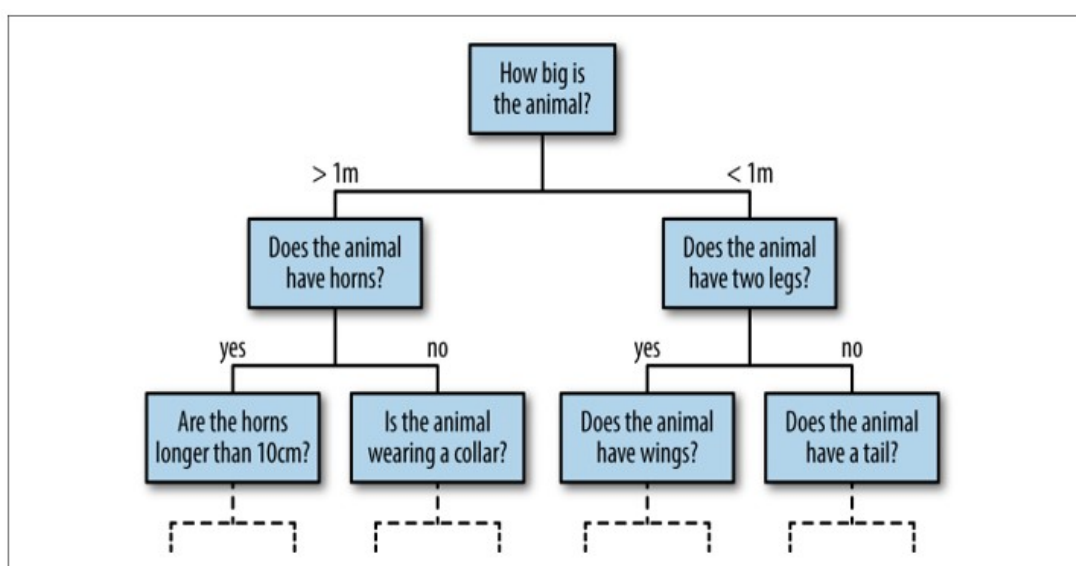


Figura 7

Ejemplo de un árbol de decisión binario.

Fuente. Tomado de VanderPlas (2022).

La eficacia de los árboles de decisión radica en su capacidad para dividir binariamente las opciones, reduciendo a la mitad el número de posibilidades con cada pregunta en un árbol bien estructurado, lo que permite afinar rápidamente las opciones incluso con una amplia gama de clases. La habilidad para elegir las preguntas adecuadas en cada paso es crucial. En las implementaciones de aprendizaje automático, estas preguntas suelen manifestarse como divisiones alineadas con los ejes en los datos, donde cada

nodo del árbol separa los datos en dos grupos utilizando un valor límite dentro de una de las características.

Sobreajuste de los Árboles de Decisión

El sobreajuste es un fenómeno frecuente en los árboles de decisión; es fácil que el árbol se vuelva demasiado complejo, adaptándose en exceso a los detalles específicos del conjunto de datos en lugar de capturar patrones generales de las distribuciones subyacentes. Una forma de identificar este sobreajuste es analizando modelos entrenados en distintos subconjuntos de datos. Por ejemplo, en la Figura 8 se muestran dos árboles entrenados, cada uno con la mitad de los datos originales.

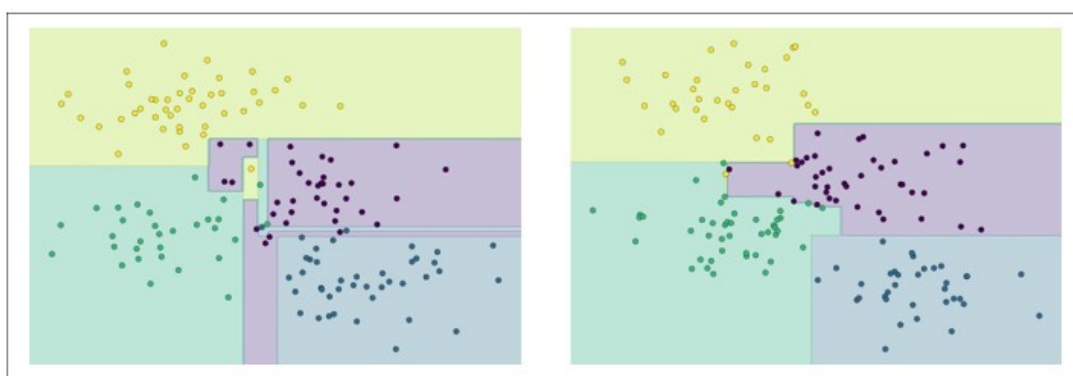


Figura 8

Ejemplo de dos árboles de decisión aleatorizados.

Fuente. Tomado de VanderPlas (2022).

Es claro que, aunque ambos árboles producen resultados coherentes en algunas áreas (como en las cuatro esquinas), en otras muestran clasificaciones significativamente diferentes (por ejemplo, en las regiones entre dos clústeres). La observación principal es que las discrepancias tienden a ocurrir en áreas donde la clasificación es más incierta. Por lo tanto, combinar la información de ambos árboles podría llevar a un resultado más preciso.

Del mismo modo que la información proveniente de dos árboles mejora nuestros resultados, es razonable esperar que utilizar la información de un mayor número de árboles podría mejorar aún más nuestros resultados.

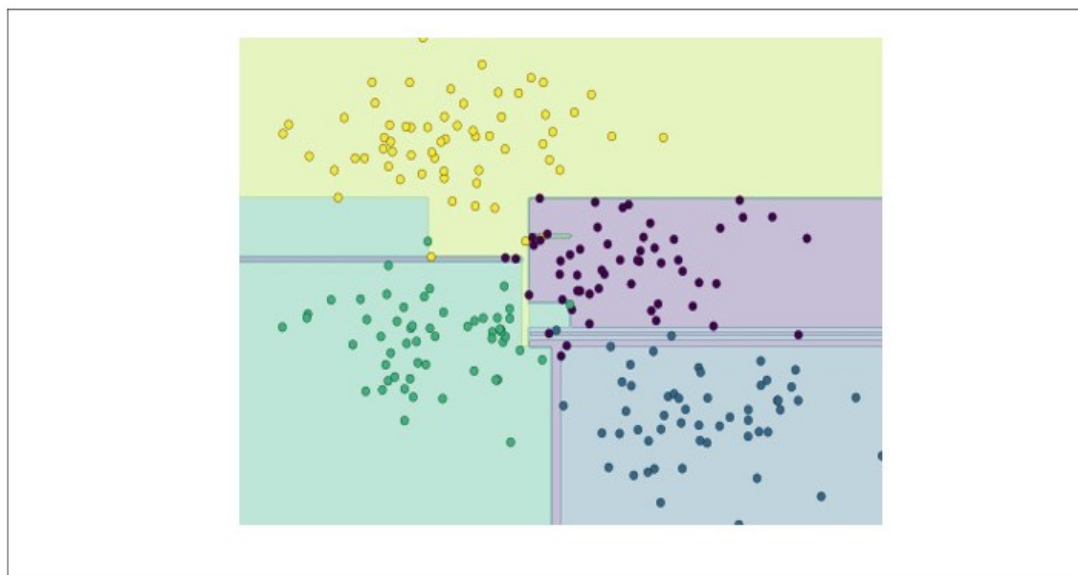


Figura 9

Ejemplo de árbol de decisión aleatorizado.

Fuente. Tomado de VanderPlas (2022).

2.4.6. Random Forests

Random Forests, constituyen un algoritmo no paramétrico de ensamble poderoso, destacado por su capacidad para combinar los resultados de una colección de estimadores simples, en este caso, árboles de decisión aleatorizados. Esta técnica se basa en el principio de que un conjunto de modelos que podría sobreajustar individualmente, puede, en conjunto, generar predicciones más precisas y robustas según VanderPlas (2022).

La eficacia de los Random Forests reside en varias ventajas clave:

- **Eficiencia en entrenamiento y predicción:** Gracias a la simplicidad de los árboles de decisión subyacentes, tanto el entrenamiento como la predicción son procesos rápidos. Además, estas tareas se pueden paralelizar fácilmente, lo que aumenta la eficiencia del algoritmo.
- **Clasificación probabilística:** Los múltiples árboles permiten realizar una clasificación basada en la probabilidad, utilizando el voto mayoritario entre los estimadores para estimar la probabilidad de cada clase.
- **Flexibilidad del modelo:** Al ser un modelo no paramétrico, los Random Forests pueden adaptarse bien a diversas tareas, superando el desempeño de otros estimadores que podrían no ajustarse adecuadamente a los datos.

Conjuntos de Estimadores: Random Forests

La idea de combinar varios estimadores que tienden al sobreajuste para reducir este efecto es la base de un método de ensamblaje conocido como bagging. Bagging utiliza un conjunto de estimadores en paralelo, donde cada uno podría sobreajustar los datos de manera individual, pero el promedio de sus resultados mejora la clasificación de acuerdo a las ideas de VanderPlas (2022).

Esta técnica de clasificación mediante bagging se puede implementar manualmente usando el meta-estimador BaggingClassifier de Scikit-Learn en Python, como se ilustra a continuación en la Figura 10:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
tree = DecisionTreeClassifier()
bag = BaggingClassifier(tree, n_estimators=100,
                        max_samples=0.8, random_state=1)
bag.fit(X, y)
```

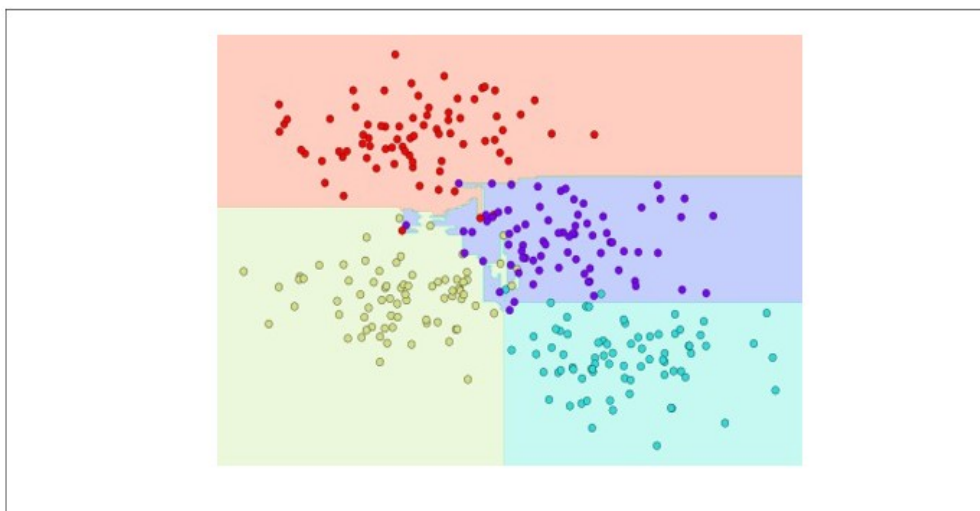


Figura 10

Ejemplo de Fronteras de decisión para un conjunto de árboles de decisión aleatorios (Random Forests).

Fuente. Tomado de VanderPlas (2022).

En este ejemplo, se ha aleatorizado los datos ajustando cada estimador con un subconjunto aleatorio del 80 % de los puntos de entrenamiento. En la práctica, los árboles de decisión se aleatorizan más efectivamente cuando se introduce cierta estocasticidad en la selección de las divisiones, permitiendo que todos los datos contribuyan en cada ajuste, pero manteniendo el nivel deseado de aleatoriedad en los resultados. Por ejemplo, al decidir sobre qué característica dividir, el árbol aleatorizado podría seleccionar entre varias de las características más significativas.

Regresión con Random Forests

En la sección anterior, exploramos los bosques aleatorios en el ámbito de la clasificación. Sin embargo, estos también pueden aplicarse efectivamente a problemas de regresión, es decir, con variables continuas en lugar de categóricas de VanderPlas (2022).

Tomemos como ejemplo el siguiente conjunto de datos, derivado de la combinación de oscilaciones rápidas y lentas:

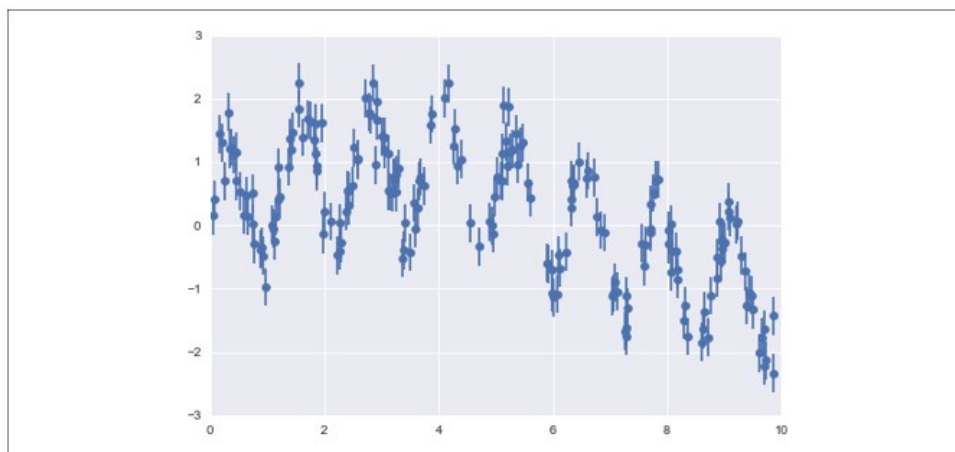


Figura 11

Ejemplo de Datos para regresión con Random Forests.

Fuente. Tomado de VanderPlas (2022).

Como se puede observar, el modelo de bosque aleatorio, al ser no paramétrico, demuestra una flexibilidad suficiente para ajustarse a datos que presentan múltiples períodos, sin la necesidad de definir explícitamente un modelo que abarque varios períodos.

2.4.7. Regresión Logística

La **regresión logística** es un modelo estadístico ampliamente utilizado para predecir resultados binarios (por ejemplo, éxito o fracaso) a partir de una o varias variables independientes. En este modelo, la probabilidad de que ocurra el evento de interés (llamado "éxito") se estima mediante la función logística, definida como:

$$p(y = 1|x; \beta) = \frac{1}{1 + e^{-\beta^T x}}$$

donde:

- $p(y = 1|x; \beta)$ representa la probabilidad de éxito dada una observación x ,
- x es un vector de variables predictoras o covariables,

- β es el vector de coeficientes que se estiman para maximizar la verosimilitud del modelo.

El objetivo en la regresión logística es encontrar los valores de β que mejor ajusten los datos observados. Para ello emplea un método de optimización iterativa como el *método de Newton-Raphson* que actualiza los valores de β mediante la siguiente expresión:

$$\beta^{(i+1)} = \beta^{(i)} - H^{-1}(\beta^{(i)}) \cdot g(\beta^{(i)})$$

donde:

- $\beta^{(i)}$ representa el valor de los coeficientes en la iteración actual i ,
- $\beta^{(i+1)}$ es el valor de los coeficientes en la siguiente iteración $i + 1$,
- $H(\beta^{(i)})$ es la matriz Hessiana, que contiene las segundas derivadas de la función de verosimilitud con respecto a β , evaluada en $\beta^{(i)}$,
- $g(\beta^{(i)})$ es el gradiente de la función de verosimilitud evaluado en $\beta^{(i)}$.

A veces se utiliza la **regresión logística regularizada**, que añade un término de regularización para mejorar la generalización del modelo y evitar el sobreajuste. Un ejemplo es la regularización ℓ_2 , en la que se incluye un término adicional $\lambda \|\beta\|^2$ en la función objetivo, donde λ es el parámetro de regularización; según lo expresado por Deza y Atamtürk (2022).

2.5. Glosario

Reclamo. Es el mecanismo de participación de la ciudadanía a través del cual las personas expresan su insatisfacción o disconformidad ante la entidad de la administración pública que lo atendió o prestó el bien o servicio. (PCM, 2021)

NLP (Procesamiento de Lenguaje Natural). Campo de la inteligencia artificial que se ocupa de la interacción entre computadoras y el lenguaje humano. Se usa para comprender, interpretar y generar lenguaje natural en aplicaciones como la clasificación de textos o el análisis de sentimientos.

ML (Aprendizaje Automático). Rama de la inteligencia artificial que crea algoritmos capaces de aprender y mejorar a partir de datos, sin ser programados explícitamente para cada tarea. Incluye técnicas como el aprendizaje supervisado, no supervisado y por refuerzo.

SVM (Máquinas de Vectores de Soporte). Algoritmo de aprendizaje supervisado que se utiliza principalmente para clasificación y regresión, destacándose por su capacidad para encontrar un límite óptimo entre clases.

TF-IDF Vectorizer. El TF-IDF Vectorizer (Term Frequency–Inverse Document Frequency Vectorizer) es una técnica de procesamiento de lenguaje natural que convierte textos en vectores numéricos ponderando la frecuencia de cada palabra en un documento y su rareza en el conjunto total. Permite resaltar los términos más relevantes y reducir la influencia de los más comunes.

Random Forests (Bosques Aleatorios). Algoritmo de ensamble que utiliza múltiples árboles de decisión para mejorar la precisión de las predicciones y evitar el sobreajuste.

Capítulo III

Metodología

3.1. Tipo y Diseño de la Investigación

3.1.1. Nivel de Investigación

Es una investigación aplicada. Se usan técnicas avanzadas de *Machine Learning* (ML) y *Procesamiento de Lenguaje Natural* (NLP) para categorizar automáticamente los reclamos de un libro de reclamaciones virtual. De esta forma mejorar la Gestión de Tecnologías de la Información y Comunicaciones.

3.1.2. Tipo de Investigación

La investigación se enfoca en el área de ciencias de la computación, específicamente en minería de datos y procesamiento de lenguaje natural (PLN).

3.1.3. Diseño de la Investigación

Experimentación con modelos para identificar si ofrecen un buen rendimiento en términos de las métricas de evaluación seleccionadas.

3.1.4. Método de la investigación.

El enfoque de la investigación será fundamentado en la aplicación de métodos como técnicas de inteligencia artificial y técnicas de NLP, para validar las hipótesis a través de la experimentación.

3.2. Dataset y Partición

3.2.1. Dataset

El dataset empleado en esta investigación corresponde a un conjunto de datos reales que recopila registros de reclamos procedentes de una plataforma virtual de gestión de reclamos. En total, contiene 261,628 registros individuales, cada uno representando un reclamo específico presentado por los usuarios. Cada registro incluye principalmente dos columnas: la **Descripción**, que contiene el texto del reclamo redactado en lenguaje natural, y el **Tipo de Registro**, que clasifica el reclamo en categorías como *Queja*, *Denuncia*, *Reclamo*, *Consulta*, *Sugerencia* u *Otros*. La primera columna constituye la variable independiente utilizada en el análisis textual mediante técnicas de *Procesamiento de Lenguaje Natural* (NLP), mientras que la segunda se emplea como variable dependiente para el proceso de clasificación con algoritmos de *Machine Learning* (ML).

El dataset¹ fue obtenido de una plataforma digital especializada en la recepción y gestión de reclamos en formato electrónico, seleccionándose por su relevancia para el objetivo de la investigación, centrado en optimizar la clasificación automática de registros mediante técnicas computacionales.

3.2.2. Partición del Dataset

El dataset, compuesto por 261,627 registros no nulos, fue dividido en dos subconjuntos siguiendo una proporción de 80 %-20 %. En total, 209,302 registros fueron utilizados para el entrenamiento del modelo, permitiendo identificar patrones y relaciones entre el texto de los reclamos y sus categorías, mientras que 52,325 registros se destinaron a la validación, con el fin de evaluar el rendimiento y la capacidad de generalización del modelo frente a datos no vistos. Esta partición asegura un equilibrio adecuado entre aprendizaje y evaluación, contribuyendo a obtener resultados más confiables y representativos.

3.2.3. Técnica de muestreo de partición

Para dividir adecuadamente el conjunto de datos en subconjuntos de entrenamiento, validación y prueba, se empleó una técnica de muestreo aleatorio estratificado. Esta técnica garantiza que la proporción de clases en cada subconjunto sea representativa del total, evitando así que alguna clase quede sobrerrepresentada o subrepresentada en una sola partición.

¹Fuente de origen: No se puede dar a conocer el nombre de la fuente del dataset por motivos de confidencialidad de la entidad.

3.3. Operacionalización de Variables

3.3.1. Identificación de Variables

Variable Independiente: Descripción del reclamo

- **Definición conceptual:** Es una secuencia de caracteres o palabras que transmite información. En este contexto, dicha información detalla el reclamo, la queja, sugerencia, denuncia, reclamo o consulta presentada por el usuario.
- **Definición operacional:** La descripción de la reclamo, después de ser sometida a un preprocesamiento que incluye la eliminación de caracteres especiales, conversión a minúsculas, eliminación de palabras vacías (*stop words*), lematización, se transforma en un vector numérico utilizando la técnica (*TF-IDF Vectorizer*). Cada elemento del vector representa la frecuencia de aparición de una palabra en el vocabulario del corpus.
- **Dimensión:** Contenido de la reclamo.

Variable Dependiente: Tipo de Registro de reclamo

- **Definición conceptual:** La variable dependiente es de naturaleza categórica y representa cómo se clasifica o etiqueta cada registro de reclamación, considerando su naturaleza y contenido.
- **Definición operacional:** Se consideran cinco categorías principales para clasificar las reclamaciones: RECLAMO, DENUNCIA, SUGERENCIA, CONSULTA, QUEJA y OTRO. Cada reclamación será asignada a una de estas categorías de acuerdo con su naturaleza específica.
- **Dimensiones:** Las dimensiones evaluadas serán:
 - Precisión
 - Sensibilidad
 - F1-score
 - Soporte

3.3.2. Matriz de Operacionalización de Variables

Tabla 2*Matriz de operacionalización de variables.*

Variable	Tipo de Variable	Definición conceptual	Definición de operacionalidad	Dimensiones	Indicadores
Descripción del reclamo.	Independiente.	Es una secuencia de caracteres o palabras que transmite información. En este contexto, dicha información detalla el reclamo, la queja, sugerencia, denuncia, reclamo o consulta presentada por el usuario.	La descripción de la reclamo, después de ser sometida a un preprocesamiento que incluye la eliminación de caracteres especiales, conversión a minúsculas, eliminación de palabras vacías (<i>stop words</i>), se transforma en un vector numérico utilizando la técnica (<i>Count-Vectorizer</i>). Cada elemento del vector representa la frecuencia de aparición de una palabra en el vocabulario del corpus.	Contenido del reclamo.	Palabras clave y frases relevantes. Temas y asuntos mencionados. Sentimiento y tono emocional del texto.

Tipo de registro de reclamo.	Dependiente.	La variable dependiente es de naturaleza categórica y representa cómo se clasifica o etiqueta cada registro de reclamación, considerando su naturaleza y contenido.	Se consideran cinco categorías principales para clasificar las reclamaciones: RECLAMO, DENUNCIA, SUGERENCIA, CONSULTA, QUEJA y OTRO. Cada reclamación será asignada a una de estas categorías de acuerdo con su naturaleza específica.	Precisión. Del inglés “Precision”, Esta métrica indica la proporción de casos positivos detectados.
				Sensibilidad. Del inglés “Recall” o “Sensitivity”, esta métrica representa la proporción de casos positivos que fueron correctamente identificados por el modelo.
				F1-score. Se trata de una métrica que combina la precisión y la sensibilidad en una sola medida de manera armónica.

Soporte.

Del inglés “Support”.Indica el número de muestras de cada clase en el conjunto de datos de prueba. Proporciona contexto para interpretar las otras métricas, ya que un soporte bajo puede hacer que las métricas sean menos fiables.

3.3.3. Descripción de Indicadores de Variables

- **Precisión:** Esta métrica indica la proporción de casos positivos correctamente identificados.
- **Sensibilidad:** Esta métrica representa la proporción de casos positivos que fueron correctamente identificados por el modelo.
- **F1-score:** Combina las métricas de precisión y sensibilidad en una sola medida armónica.
- **Soporte:** Indica el número de muestras de cada clase en el conjunto de datos de prueba, proporcionando contexto para interpretar las otras métricas.

3.4. Justificación de los Algoritmos de Clasificación

La elección de los algoritmos de clasificación en esta investigación no es arbitraria ni meramente experimental, sino que se encuentra sólidamente fundamentada en evidencia empírica acumulada y en criterios científicos derivados de la literatura especializada en aprendizaje automático aplicado al procesamiento de lenguaje natural (NLP) en contextos institucionales. A partir del análisis de antecedentes revisados sistemáticamente, se identificaron patrones recurrentes en el uso de algoritmos clásicos de clasificación como *Naive Bayes*, *Random Forest*, *Regresión Logística* y *Support Vector Machines (SVM)*, todos ellos destacados por su efectividad en tareas de categorización multiclase de textos ciudadanos. Cabe precisar que, si bien muchas de estas investigaciones se han desarrollado sobre corpus en inglés u otros idiomas extranjeros, aún se requieren más estudios aplicados directamente a textos institucionales en español, que incorporen sus particularidades léxicas, sintácticas y semánticas.

Por ejemplo, Bozyiğit et al. (2022) diseñaron un sistema para clasificar reclamos en el sector público alimentario, utilizando un corpus real y algoritmos como *Naive Bayes*, *SVM*, *Random Forest*, y *Regresión Logística*. Los modelos lograron métricas de F1-score superiores al 80 % cuando fueron combinados con técnicas de representación TF-IDF y selección de atributos mediante Chi-cuadrado, lo que evidencia la capacidad de estos clasificadores para manejar datos textuales con alto nivel de ruido y variabilidad lingüística.

Saha et al. (2024), por su parte, implementaron un sistema integral de clasificación y enrutamiento de reclamos públicos que también empleó *Random Forest*, *SVM*, *Regresión Logística* y modelos basados en *BERT*. Los resultados de este estudio mostraron mejoras operativas y reducciones significativas de la carga manual gracias a la alta

precisión alcanzada por estos modelos, incluso en entornos multilingües y multicanal. Esta evidencia refuerza la aplicabilidad de modelos supervisados tradicionales en contextos gubernamentales digitalizados.

En un contexto más empresarial, Peixoto (2021) exploró la clasificación automática de reclamos en plataformas de servicio al cliente usando algoritmos como *Naive Bayes*, *SVM*, redes neuronales (MLP) y *DistilBERT*, encontrando que los modelos clásicos —bien ajustados— ofrecían un rendimiento competitivo frente a alternativas más complejas, siendo además más eficientes computacionalmente y más interpretables, dos atributos esenciales para su implementación en plataformas públicas.

Asimismo, Naik et al. (2023) realizaron un estudio comparativo donde modelos de *SVM* lineal y *Regresión Logística* obtuvieron una exactitud superior al 85 % usando representaciones bigram sobre reclamos financieros. Su análisis evidenció que, en tareas de clasificación textual con estructuras bien definidas, estos modelos ofrecen un equilibrio ideal entre rendimiento y eficiencia.

Dado este marco, la presente investigación incluye los siguientes algoritmos:

- **Naive Bayes (Multinomial):** ampliamente utilizado en clasificación de texto por su rapidez y efectividad en contextos donde la independencia entre atributos (palabras) puede ser asumida razonablemente. Se justifica también por su bajo costo computacional y su capacidad para trabajar eficientemente con vectores TF-IDF.
- **Regresión Logística:** seleccionada por su eficacia comprobada en clasificación multiclase, su interpretabilidad y su desempeño estable incluso con datos de alta dimensionalidad, atributos que la posicionan como uno de los algoritmos más fiables en aplicaciones institucionales.
- **Random Forest:** elegido por su capacidad para modelar relaciones no lineales y manejar conjuntos de datos con ruido y desbalance de clases, así como por su robustez frente al sobreajuste y su utilidad como método de referencia en problemas de clasificación supervisada.
- **Linear Support Vector Classifier (LinearSVC):** incorporado debido a su eficacia en espacios vectoriales extensos —como los generados por técnicas de vectorización de texto— y su capacidad para maximizar márgenes de separación entre clases. Además, su escalabilidad y rapidez de entrenamiento lo convierten en una opción ideal para grandes volúmenes de datos institucionales clasificados en múltiples categorías.

- **Grandes Modelos de Lenguaje (LLM):** En el contexto de las tecnologías actuales de procesamiento del lenguaje natural, los Modelos de Lenguaje Grande (Large Language Models, LLM) constituyen una alternativa reciente para el análisis y clasificación de texto. Basados en arquitecturas de transformers, estos modelos utilizan representaciones contextuales aprendidas a partir de grandes volúmenes de datos. En esta investigación se emplea la herramienta Claude AI para la clasificación automática de reclamos ciudadanos en el sector público, con el fin de evaluar su desempeño y analizar su aplicabilidad en entornos institucionales.

La combinación de estos algoritmos responde, entonces, a una lógica de comparación metodológica robusta: se cubren distintos paradigmas de clasificación (probabilístico, lineal, de margen máximo, de ensamble y de grandes modelos de lenguaje), lo cual permite no solo contrastar sus desempeños de manera cuantitativa mediante métricas como precisión, exactitud y F1-score, sino también analizar su viabilidad práctica en escenarios reales del sector público. En suma, esta selección metodológicamente informada busca garantizar la solidez del proceso de clasificación automática y aportar evidencia útil para futuras implementaciones operativas en la gestión pública digital.

3.5. Configuración e inicialización de los algoritmos

Para la etapa de modelado se emplearon diversos algoritmos de clasificación supervisada, representativos de distintos enfoques de aprendizaje automático. En esta sección se describe la configuración adoptada y la justificación técnica de los valores asignados a sus principales hiperparámetros.

3.5.1. Naive Bayes (NB)

Se utilizó el modelo *Naive Bayes*, mediante `MultinomialNB` de `scikit-learn`, con los parámetros `alpha = 0.0098` y `fit_prior = True`. El valor de `alpha` se definió buscando un suavizado que mitigara el impacto de términos poco frecuentes sin distorsionar la distribución de clases. La opción `fit_prior = True` permitió ajustar las probabilidades a priori según la distribución del corpus, mejorando el rendimiento en escenarios desbalanceados.

3.5.2. Linear Support Vector Classifier (LinearSVC)

El modelo *LinearSVC* se configuró con `C = 1.0`, `loss = 'squared_hinge'`, `dual = False`, `max_iter = 2000` y `random_state = 42`. El valor de `C` equilibró la penalización del error y la regularización del margen, mientras que la pér-

dida cuadrática favoreció la estabilidad numérica. La opción `dual = False` mejoró la eficiencia dado que el número de muestras superaba al de características, y se amplió el número de iteraciones para garantizar convergencia en espacios de alta dimensión.

3.5.3. Random Forest (RF)

El modelo *Random Forest* se configuró con hiperparámetros `n_estimators = 100`, `min_samples_leaf = 10`, `max_features = 'sqrt'` y `random_state = 42`. El número de árboles buscó estabilidad con un costo computacional moderado, y el tamaño mínimo de hoja se incrementó ligeramente para reducir sobreajuste. La selección de `max_features = 'sqrt'` promovió la diversidad entre árboles, disminuyendo la correlación interna del ensamble.

3.5.4. Regresión Logística (LR)

La *Regresión Logística* se configuró con `solver = 'lbfgs'`, `penalty = 'l2'`, `C = 7`, `max_iter = 1000` y `random_state = 42`. El solucionador `lbfgs` ofreció eficiencia en espacios con muchas variables, mientras que la regularización `l2` evitó sobreajuste manteniendo la interpretabilidad. El valor de `C` se fijó para mejor generalización al reducir la fuerza de regularización.

3.5.5. Modelos de Lenguaje Grande (LLM) – Claude AI

Se utilizó un *Large Language Model* (LLM), el agente de *Claude AI*, basado en arquitecturas *transformer*. El modelo se utilizó en forma de clasificador con su limitante máximo de recepción de datos para clasificación siendo que solo se podía enviar unos 10000 muestras que se buscaron estuvieran correctamente estratificadas. Su funcionamiento se basó en la formulación de un *prompt* consistente que garantice estabilidad y coherencia en la clasificación textual.

Prompt de clasificación de reclamos ciudadanos

Eres un modelo de lenguaje especializado en clasificación de texto institucional. Tu tarea es analizar descripciones textuales de reclamos ciudadanos y asignar a cada una una categoría correspondiente entre las clases definidas.

Información del problema:

- **Dominio:** Gestión pública y libro de reclamaciones virtual.
- **Idioma:** Español (variedad peruana).
- **Fuente de datos:** Descripciones textuales de reclamos ingresados por ciudadanos.
- **Objetivo:** Clasificar automáticamente cada descripción en una de las siguientes seis categorías:
 1. **Reclamo** → Cuando el usuario expresa disconformidad con un servicio recibido o incumplimiento de lo ofrecido.
 2. **Queja** → Cuando el usuario manifiesta malestar o incomodidad por la atención recibida, sin pedir una solución formal.
 3. **Denuncia** → Cuando el usuario reporta una falta, infracción o acto indebido de un trabajador o área.
 4. **Consulta** → Cuando el usuario formula una pregunta o solicitud de información.
 5. **Sugerencia** → Cuando el usuario propone una mejora o recomendación.
 6. **Otro** → Cuando la descripción no encaja en las categorías anteriores o es ambigua.

Modo de operación:

1. Recibir un archivo con las descripciones a clasificar.
2. Entregar posteriormente los resultados de los casos clasificados.

Instrucciones adicionales:

- No inventar nuevas clases ni subdividir las existentes.
- Clasificar basándose en el contenido semántico, tono y propósito del texto.
- Mantener coherencia en el uso de las categorías.
- Si hay ambigüedad o texto incompleto, asignar la clase “Otro”.
- No incluir explicaciones ni comentarios en la salida, solo la categoría final por fila.

Este prompt fue diseñado para maximizar la precisión semántica del modelo al delimitar claramente las categorías y establecer un flujo de trabajo estructurado. Su propósito es estandarizar la interpretación del texto por parte del LLM, garantizando coherencia en la clasificación automática de los reclamos ciudadanos.

3.6. Procedimientos

El procedimiento de la experimentación para la validación de las hipótesis de la investigación se llevó a cabo en varias etapas en la arquitectura general del proceso. A continuación, se describen estos pasos en detalle:

3.6.1. Arquitectura General del Proceso

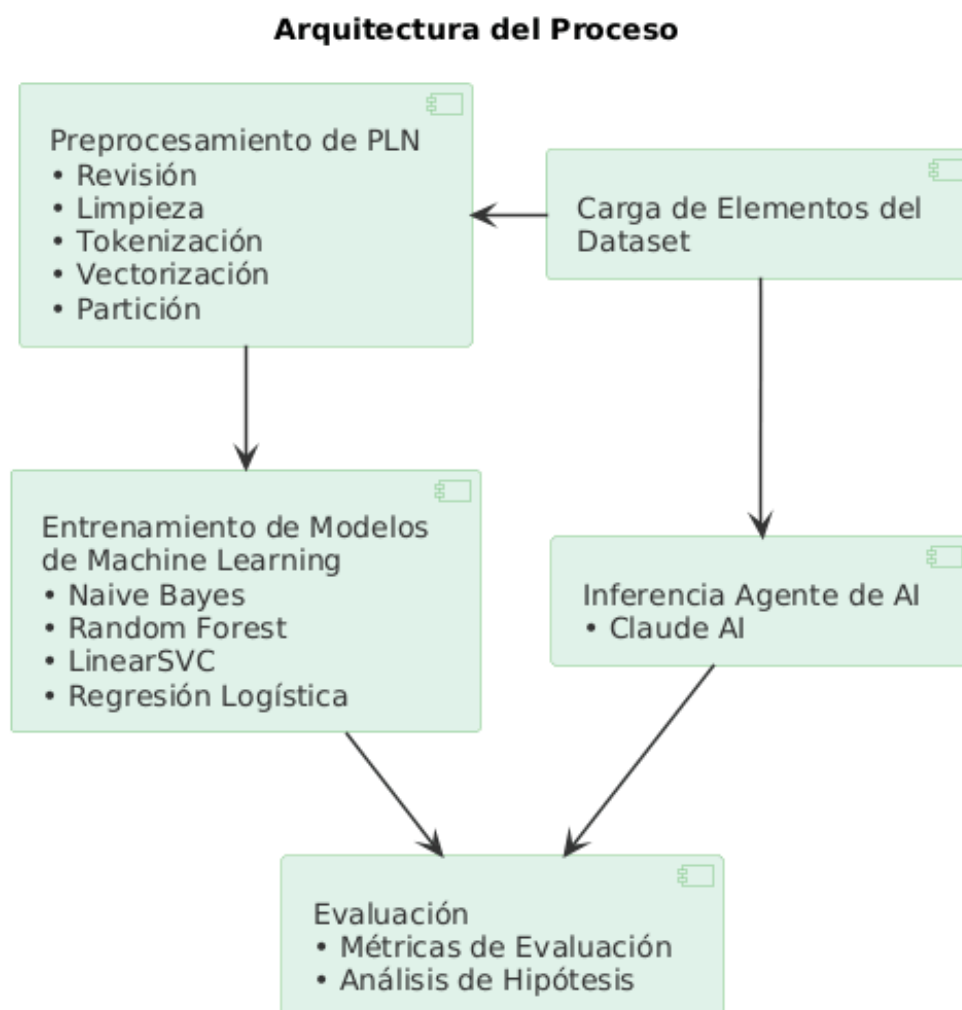


Figura 12

Arquitectura general del modelo propuesto.

3.6.2. Experimentos de Validación para las Hipótesis

En esta sección se describe el conjunto de experimentos planificados para comprobar las hipótesis específicas de la investigación. El desarrollo experimental tiene como propósito aplicar técnicas de aprendizaje automático y procesamiento de lenguaje natural (NLP) para clasificar los reclamos registrados en el libro de reclamaciones virtual, evaluando distintos algoritmos y configuraciones bajo condiciones controladas.

El proceso experimental se estructura en fases que abarcan la preparación de datos, el preprocesamiento lingüístico, la representación numérica de los textos, el entrenamiento de modelos de clasificación y la simulación del flujo operativo institucional. Cada una de estas etapas fue diseñada para garantizar la reproducibilidad de los resultados y la validez de las comparaciones entre los modelos.

1. Observación previa del dataset. Se parte del conjunto de datos denominado *Reclamos_DT*, compuesto por más de doscientas sesenta mil entradas textuales clasificadas en seis tipos de registro: Reclamo, Consulta, Otro, Queja, Denuncia y Sugerencia. Antes de proceder al entrenamiento, se realiza un análisis exploratorio para determinar la frecuencia y el equilibrio de las clases, así como la extensión promedio de las descripciones textuales. Esta revisión permite identificar desbalances significativos entre categorías y establecer la necesidad de aplicar estrategias de remuestreo o balanceo posterior.

2. Preprocesamiento con técnicas NLP. Con el fin de transformar los textos en representaciones adecuadas para los modelos de aprendizaje automático, se aplicará un proceso de preprocesamiento lingüístico estructurado en varias etapas. En primer lugar, se limpiarán los datos eliminando valores nulos o inconsistentes. Luego se estandarizarán las etiquetas de las categorías para mantener uniformidad en el conjunto de entrenamiento. Posteriormente, se realizará la tokenización y lematización de los textos utilizando la biblioteca `spaCy` y su modelo en español `es_core_news_sm`, junto con la eliminación de signos de puntuación y palabras vacías (*stopwords*). Una vez normalizado el texto, se generarán dos tipos de representaciones numéricas: una basada en *TF-IDF* y otra en *embeddings* semánticos mediante el modelo multilingüe `paraphrase-multilingual-MiniLM-L12-v2`. Estas representaciones permitirán comparar la eficacia de distintas estrategias de vectorización en la clasificación de los reclamos.

3. Entrenamiento de modelos de clasificación. Con las representaciones textuales generadas, se dividirá el conjunto de datos en subconjuntos de entrenamiento y prueba utilizando un muestreo estratificado para conservar la proporción de clases. Sobre estos datos se entrenarán diferentes modelos de aprendizaje supervisado:

- **Naive Bayes Multinomial**, configurado con parámetros ajustados para datos textuales discretos.
- **Regresión Logística**, con regularización de tipo L2 para evitar sobreajuste.
- **Linear SVC**, optimizado para espacios de alta dimensionalidad.

- **Random Forest**, usando como entrada los vectores de *embeddings* semánticos.

Cada modelo se entrenará con sus conjuntos de datos respectivos y será evaluado mediante métricas estándar de clasificación multiclase: precisión, *recall*, *support* y F1-score, por clase.

4. Evaluación y comparación de modelos. Tras el entrenamiento, se procederá a evaluar el rendimiento de cada modelo utilizando los conjuntos de prueba previamente definidos. Las métricas obtenidas permitirán comparar la capacidad de generalización, la estabilidad entre clases y la coherencia de las predicciones. El análisis comparativo considerará además las características de cada técnica, tales como el tiempo de entrenamiento, la complejidad computacional y la sensibilidad al desbalance de clases. Con base en estas evaluaciones, se identificará el modelo con mejor equilibrio entre precisión y eficiencia para la tarea de clasificación.

5. Clasificación Manual de Reclamos. De manera complementaria, se observa el flujo operativo de atención de reclamos mediante la clasificación tradicional de forma manual. Los resultados servirán como referencia para contrastar el comportamiento de un sistema automatizado frente al procedimiento manual tradicional.

3.6.3. Validación para las Hipótesis Específicas

Hipótesis específica 1. *El diseño y entrenamiento de un modelo de machine learning mejora la clasificación eficiente de reclamos en un libro de reclamaciones virtual, asegurando precisión y relevancia en las categorías utilizadas.*

- **Evaluación comparativa de modelos.** Se propone realizar una comparación cuantitativa entre los modelos entrenados (Naive Bayes, Regresión Logística, Linear SVC, Random Fores), comparando también con un Agente de AI, y su respectivo análisis. El objetivo es determinar si los modelos logran una correcta clasificación de los textos de las diferentes categorías. Las métricas se utilizarán para analizar la eficiencia del diseño y entrenamiento de cada modelo.
- **Análisis de eficiencia operativa.** Se planifica un análisis complementario de impacto, en el cual los resultados de clasificación obtenidos se integrarán a un modelo de simulación que represente el flujo de gestión de reclamos. Este experimento permitirá estimar cómo el uso de los modelos podría influir en los tiempos de atención y la priorización de los casos. De esta manera, se podrá cuantificar el efecto de los modelos de clasificación de machine learning.

Hipótesis específica 2. *La implementación de técnicas de procesamiento de lenguaje natural (NLP) en un sistema de machine learning mejora la comprensión y categorización automática de los textos de reclamos en un entorno virtual.*

- **Evaluación de la aplicación de NPL.** Se plantea realizar una comparación entre diferentes configuraciones de procesamiento lingüístico aplicadas a los modelos. Se evaluará el desempeño de los modelos que utilizan TF-IDF (*Linear SVC*, Naive Bayes y Regresión Logística), embeddings (*Random Forest*) y representaciones semánticas *LLM*. El propósito es identificar el aporte de cada técnica de NLP en la comprensión de los textos y en la precisión de la categorización automática.
- **Profundidad semántica y comprensión contextual.** Este experimento buscará determinar el impacto de incorporar modelos según su capacidad semántica, como los incorporados en TF-IDF, embeddings y representaciones semánticas. Se evaluará si las representaciones del lenguaje natural permiten una interpretación coherente y contextual de los reclamos, analizando su efecto en la consistencia de las predicciones y en la clasificación de categorías con lenguaje ambiguo o complejo. Los resultados de esta comparación servirán para validar la relación directa entre la profundidad semántica del modelo y la mejora en la categorización automática.

Capítulo IV

Fase Experimental

4.1. Observación previa del dataset *Reclamos_DT*.

En la fase de análisis exploratorio inicial, se examinó la variable *Tipo_registro* con el fin de identificar las distintas categorías presentes en el conjunto de datos original, previo a cualquier proceso de estandarización o depuración semántica. Los resultados mostraron una amplia diversidad de etiquetas, con un total de 694 categorías diferentes y frecuencias muy variables entre sí.

La distribución de los tipos de registro se resume en la Tabla 3, donde se observa una marcada concentración en pocas categorías principales, tales como *reclamo* (189 335 registros), *consulta* (25 730 registros), *Otro:* (23 410 registros), *queja* (10 870 registros) y *denuncia* (1 806 registros). Sin embargo, el conjunto también incluye una gran cantidad de categorías con una frecuencia mínima, muchas de ellas con un solo registro.

Tabla 3

Distribución completa de la variable Tipo de registro

Tipo de registro	Frecuencia
Reclamo	189 335
Consulta	25 730
Otro:	23 410
Queja	10 870
Denuncia	1 806
⋮	⋮
Otro: APRECIACIÓN	1
Otro: Es para otra entidad	1
Otro: [PERSONA_74005]	1
Otro: [LUGAR_7089]	1
Total de categorías distintas	694

El análisis de esta tabla evidencia que la variable *Tipo_registro* presenta un alto grado de dispersión y redundancia en las etiquetas, derivado de la ausencia de un esquema de clasificación uniforme. Esta característica justifica la necesidad de un proceso posterior de estandarización de etiquetas, que permita agrupar las categorías semánticamente equivalentes y reducir la dimensionalidad del conjunto de datos.

4.2. Privacidad y confidencialidad de los datos

Se implementaron medidas de anonimización sobre los datos sensibles contenidos en el dataset utilizado, con el fin de garantizar la confidencialidad de la información.

4.3. Preprocesamiento de NPL

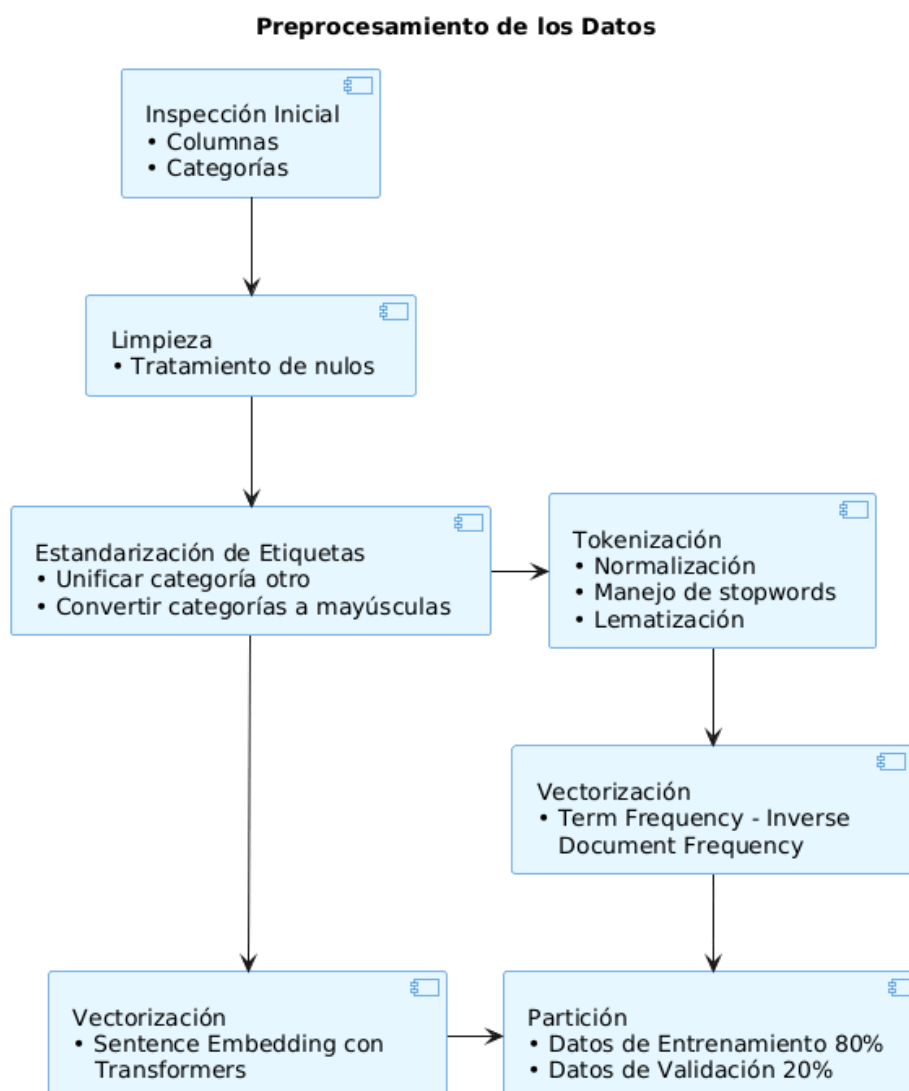


Figura 13

Preprocesamiento del Dataset.

4.3.1. Inspección Inicial

Se realizó una verificación de valores nulos en el conjunto de datos original, el cual comprendía un total de 261 628 registros y 2 variables.

Tabla 4

Estructura inicial del conjunto de datos

Descripción	Cantidad
Número de registros (filas)	261 628
Número de variables (columnas)	2
Dimensión total del conjunto	(261 628, 2)

4.3.2. Limpieza del dataset

La detección de valores nulos se muestra en la Tabla 5, donde se observa que únicamente una observación carecía de información en la columna *Descripción*. A fin de preservar la consistencia e integridad del conjunto de datos, se procedió a eliminar dicho registro. Como resultado, el conjunto final quedó conformado por 261 627 registros y las mismas 2 variables, garantizando la ausencia total de datos faltantes para las fases posteriores de análisis.

Tabla 5

Verificación de valores nulos en el conjunto de datos original

Variable	Valores nulos	Valores válidos
Descripción	1	261 627
Tipo_registro	0	261 628

Después de esta depuración, el tamaño del conjunto de datos pasó de 261 628 a 261 627 registros, asegurando que todas las observaciones incluyan información completa en ambas variables analizadas.

4.3.3. Estandarización de Etiquetas

Como parte del proceso de preparación de los datos, se efectuó la estandarización de la variable *Tipo_registro*, cuyo propósito fue unificar las diversas etiquetas que representaban categorías equivalentes. En la etapa exploratoria previa, esta variable contenía 694 categorías distintas, muchas de las cuales correspondían a variaciones textuales, errores de digitación o descripciones particulares ingresadas por los usuarios.

Para garantizar la coherencia semántica, se aplicaron transformaciones de limpieza y normalización, incluyendo la conversión de mayúsculas/minúsculas, la eliminación de

espacios en blanco, y la agrupación de todas las etiquetas que iniciaban con la palabra “Otro” o “Acceso” dentro de una única categoría denominada *OTRO*. De este modo, se redujo la dispersión de valores y se consolidaron las etiquetas en seis categorías principales, que se muestran en la Tabla 6.

Tabla 6

Distribución de la variable Registro tras la estandarización de etiquetas

Tipo de Registro	Frecuencia
RECLAMO	189 334
OTRO	32 921
CONSULTA	25 730
QUEJA	10 870
DENUNCIA	1 806
SUGERENCIA	966
Total de registros	261 627

Tabla 7

Registros clasificados por su categoría

Nº	Descripción	Registro
1	Estimado señor:\nPor medio de la presente, qui...	SUGERENCIA
2	Mercadería mojada o deteriorada y mal trato d...	SUGERENCIA
3	No me llaman dándome la clave, para elegir mi ...	RECLAMO
4	Buenas noches vengo en reiteradas veces quere...	SUGERENCIA
5	[LUGAR_005] (número de pedido) Empresa: Crea...	SUGERENCIA
6	No recibía el código para continuar el procedim...	RECLAMO
...		
261623	Revise mi solicitud de [PERSONA_317] de papele...	RECLAMO
261624	Solicito mi cita de recojo ya que a través de ...	RECLAMO
261625	En fecha 30-01-2024, asistí a mi cita para dat...	RECLAMO
261626	[PERSONA_1541].\n[LUGAR_1137] es respecto a l...	RECLAMO
261627	En fecha 16-12-2023, inicié mi trámite de CAMB...	RECLAMO

El resultado de esta depuración permitió disponer de una variable categórica más homogénea y útil, reduciendo significativamente la complejidad inicial del conjunto de datos. Sin embargo, se identificó un marcado **desbalanceo entre las categorías**, siendo *RECLAMO* la clase predominante y *SUGERENCIA* la de menor representación.

Con el fin de mitigar este desequilibrio y evitar sesgos en los modelos de aprendizaje supervisado, se aplicará el método **SMOTE** (Synthetic Minority Over-sampling Technique), el cual genera muestras sintéticas de las clases minoritarias para obtener una distribución balanceada entre las categorías antes del entrenamiento de los modelos predictivos.

4.3.4. Tokenización y Vectorización TF-IDF

Una vez estandarizadas las etiquetas categóricas, se desarrolló el proceso de tratamiento lingüístico de la variable *Descripción*, con el objetivo de preparar los textos para su posterior análisis mediante modelos de aprendizaje automático. Esta etapa se llevó a cabo utilizando la biblioteca `spaCy` y el modelo `es_core_news_md`, especializado en el procesamiento del idioma español, lo que permitió una depuración morfosintáctica precisa y coherente con las estructuras lingüísticas del corpus.

El procedimiento comprendió las siguientes etapas:

- **Tokenización:** cada descripción textual fue segmentada en unidades léxicas denominadas *tokens*, que corresponden a palabras o signos reconocidos por el modelo lingüístico. Esta segmentación permitió aislar cada elemento léxico para su análisis posterior.
- **Lematización:** se aplicó un proceso de reducción de cada palabra a su forma canónica o *lema*, unificando así las variaciones morfológicas de un mismo término (por ejemplo, “atendiendo”, “atendió” y “atender” se normalizan como “atender”). Este paso permitió reducir la dimensionalidad del vocabulario sin perder información semántica relevante.
- **Eliminación de stopwords:** se eliminaron las palabras funcionales del español (como “el”, “la”, “de”, “que”), al no aportar valor semántico para el análisis. De este modo, el texto se concentró en los términos con contenido informativo significativo.
- **Normalización y limpieza:** se aplicaron transformaciones adicionales para homogeneizar el texto, incluyendo la conversión de todos los caracteres a minúsculas, la eliminación de signos de puntuación, espacios en blanco, números y otros caracteres no alfabéticos. Este conjunto de operaciones garantizó un corpus limpio, uniforme y analíticamente consistente.

Como resultado de este proceso, cada descripción original fue convertida en una versión depurada compuesta únicamente por los lemas significativos. El corpus final, integrado por 261 627 observaciones procesadas, sirvió como base para la representación numérica mediante la técnica de **Term Frequency–Inverse Document Frequency (TF–IDF)**.

La vectorización se implementó utilizando el módulo `TfidfVectorizer` de la biblioteca `scikit-learn`, el cual transforma el texto en una matriz numérica ponderando cada término de acuerdo con su relevancia dentro del conjunto de documen-

tos. El modelo fue configurado con los siguientes parámetros: `ngram_range = (1, 2)` para incorporar unigramas y bigramas; `sublinear_tf=True` para aplicar una escala logarítmica sobre las frecuencias de los términos; `min_df = 5` y `max_df = 0.9` para eliminar términos extremadamente raros o excesivamente frecuentes; y `max_features = 50 000` para limitar el tamaño del vocabulario y optimizar la eficiencia computacional.

El resultado fue una matriz dispersa de dimensión (261 627, 50 000), con un total de 29.45 millones de elementos almacenados. En esta estructura, cada fila representa un documento individual y cada columna un término del vocabulario, ponderado por su valor TF-IDF.

4.3.5. Vectorización Sentence Embedding con Transformers

Con el propósito de obtener una representación semántica más profunda del contenido textual, se implementó un proceso de vectorización basado en **modelos de lenguaje preentrenados tipo Transformer**, específicamente mediante la generación de *sentence embeddings*. Este enfoque permitió capturar no solo la frecuencia de los términos, sino también las relaciones contextuales y significados latentes entre las palabras en cada descripción.

Para ello, se utilizó la biblioteca `Sentence-Transformers` junto con el modelo multilingüe “**paraphrase-multilingual-MiniLM-L12-v2**”, una arquitectura optimizada para la generación eficiente de representaciones vectoriales de frases en múltiples idiomas, incluido el español. Este modelo, derivado de la familia *MiniLM*, mantiene un equilibrio entre rendimiento computacional y calidad semántica, generando embeddings de dimensión 384 con un alto grado de coherencia contextual.

El procedimiento se desarrolló de la siguiente manera:

- Se cargó el modelo preentrenado en entorno CUDA para aprovechar la aceleración mediante GPU, mejorando significativamente el tiempo de procesamiento.
- El corpus total, compuesto por 261 627 descripciones textuales, fue dividido en seis subconjuntos de aproximadamente 50 000 registros cada uno, con el fin de optimizar la gestión de memoria durante la codificación masiva.
- Cada subconjunto fue procesado mediante el método `encode()` del modelo, con un tamaño de lote de 128 elementos (`batch_size=128`), generando para cada texto un vector denso de 384 características.

- Al finalizar cada iteración, los embeddings fueron almacenados individualmente en archivos binarios (`.npy`) para su posterior consolidación.
- Finalmente, los lotes fueron cargados y concatenados utilizando la biblioteca `NumPy`, conformando una matriz final de dimensiones (261 627, 384).

La estructura resultante es una matriz densa donde cada fila representa una descripción textual y cada columna una dimensión latente del espacio semántico aprendido por el modelo. A diferencia de los métodos clásicos como TF-IDF, los *sentence embeddings* capturan dependencias contextuales, sinónimos y relaciones sintácticas, lo que permite una representación más rica del contenido textual.

El resultado de esta fase fue una representación vectorial semánticamente informada del corpus textual, que captura patrones contextuales y temáticos imposibles de identificar mediante enfoques de frecuencia. Estos embeddings constituyeron una alternativa avanzada a la vectorización TF-IDF, y serán empleados como insumo en el entrenamiento del modelo de aprendizaje supervisado para la clasificación de registros textuales como el *Random Forest*.

Un extracto simplificado de la estructura de la matriz se presenta en la Tabla 8. Aunque los valores individuales no poseen interpretación directa, en conjunto describen la representación distribuida de los textos, permitiendo a los algoritmos de aprendizaje supervisado identificar similitudes y diferencias semánticas entre las observaciones.

Tabla 8

Fragmento ilustrativo de la matriz de embeddings

Dim. 1	Dim. 2	Dim. 3	...	Dim. 383	Dim. 384
-0.0757	0.1360	-0.1388	...	-0.1672	0.0363
-0.0004	-0.0166	-0.0274	...	0.2465	0.0409
0.1002	0.1088	0.0696	...	-0.2869	-0.1042
0.0528	0.2795	-0.0633	...	-0.1218	-0.0525
-0.0958	-0.0444	-0.0469	...	0.0856	0.0606

La matriz completa tuvo un total de $261\,627 \times 384 = 100\,364\,768$ valores numéricos.

4.3.6. Partición

Con el objetivo de evaluar el desempeño de los modelos supervisados de clasificación de manera objetiva, se efectuó una partición del conjunto de datos en subconjuntos de **entrenamiento** y **prueba**. Esta división permite garantizar la validez del proceso de aprendizaje y la generalización de los resultados obtenidos.

Para mantener la representatividad de las categorías de la variable dependiente *Registro*, se aplicó una **estratificación** durante la partición, asegurando que la proporción de cada clase se mantuviera constante en ambos subconjuntos. En todos los casos, se empleó una proporción del 80 % para el entrenamiento y del 20 % para la evaluación.

El proceso se realizó de manera independiente para cada forma de representación vectorial, conforme a lo siguiente:

- **Partición del conjunto TF-IDF:** a partir de la matriz dispersa generada por la vectorización TF-IDF (dimensión (261 627, 50 000)), se obtuvo un conjunto de entrenamiento de 209 301 registros y un conjunto de prueba de 52 326 registros.
- **Partición del conjunto de embeddings semánticos:** para la matriz densa de *sentence embeddings* de dimensión (261 627, 384), se aplicó la misma proporción y método de estratificación, resultando también en 209 301 observaciones para el entrenamiento y 52 326 para la prueba. Este conjunto conserva la distribución original de clases de la variable dependiente.

Tabla 9

Distribución de los conjuntos de datos por método de vectorización

Representación	Dimensiones originales	Entrenamiento (80 %)	Prueba (20 %)
TF-IDF	(261 627, 50 000)	(209 301, 50 000)	(52 326, 50 000)
Sentence Embeddings	(261 627, 384)	(209 301, 384)	(52 326, 384)

Esta estructura de partición permite que los modelos sean entrenados sobre la mayoría de los datos disponibles, mientras que una fracción independiente y no vista durante el aprendizaje se utiliza para la evaluación del rendimiento. De esta manera, se garantiza una comparación justa entre los enfoques basados en frecuencia (TF-IDF) y aquellos basados en significado contextual (embeddings Transformer), permitiendo analizar su capacidad de generalización en el proceso de clasificación de registros textuales.

4.4. Entrenamiento de los Modelos

Para la realización de las pruebas se siguió el siguiente proceso:

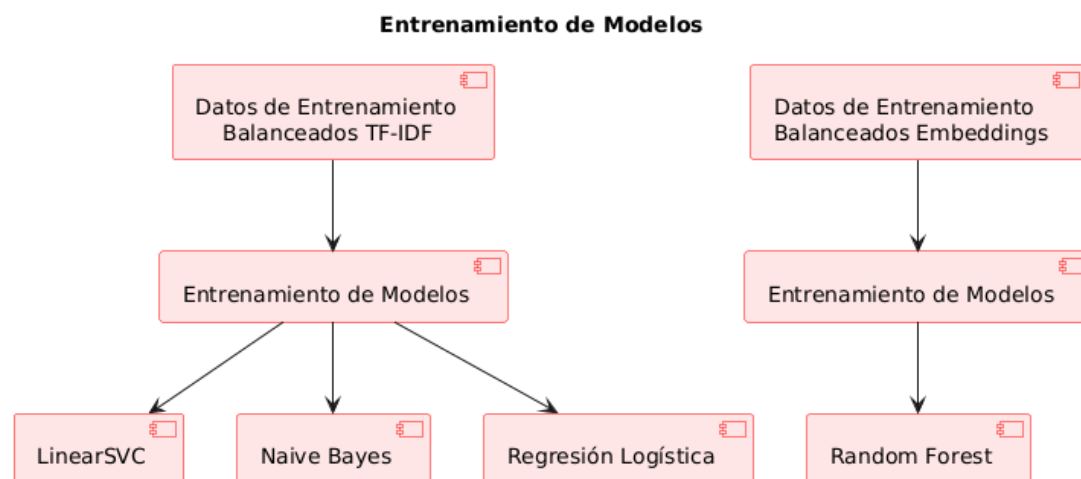


Figura 14
Entrenamiento de los Modelos.

4.5. Balanceo de Datos

Durante la fase de preparación de los conjuntos de entrenamiento, se aplicó un proceso de balanceo de clases mediante la técnica **SMOTE (Synthetic Minority Over-sampling Technique)** con el propósito de mitigar el sesgo causado por la desproporción entre categorías. El balanceo se implementó de manera independiente sobre ambos conjuntos de características —tanto el derivado de la vectorización **TF-IDF** como el correspondiente a los **embeddings**—, garantizando así una representación equitativa de las etiquetas en el proceso de entrenamiento y favoreciendo la estabilidad de los modelos en la fase de validación.

4.6. Experimentación para las Hipótesis

4.6.1. Naive Bayes (alpha = 0.0098, fit_prior = True)

Tabla 10
Reporte de clasificación en validación - Naive Bayes

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.3877	0.4118	0.3994	5146
DENUNCIA	0.1858	0.1524	0.1674	361
OTRO	0.3788	0.2443	0.2971	6585
QUEJA	0.5421	0.3671	0.4377	2174
RECLAMO	0.8119	0.8737	0.8417	37867
SUGERENCIA	0.0753	0.0363	0.0490	193

Tiempo de Entrenamiento: 0.97 segundos.

El modelo **Naive Bayes Multinomial** mostró un desempeño heterogéneo en la clasificación de textos. La categoría **“RECLAMO”** alcanzó los mejores resultados (precisión = 0.8119, recall = 0.8737, F1 = 0.8417), evidenciando una detección sólida, mientras que **“DENUNCIA”** y **“SUGERENCIA”** presentaron valores bajos de F1 (0.1674 y 0.0490, respectivamente). Este contraste se debe principalmente al **desequilibrio de clases**, ya que **“RECLAMO”** posee una cantidad muy superior de registros frente a las categorías minoritarias, y a la similitud léxica entre etiquetas como *queja* o *consulta*. Se empleó el parámetro `fit_prior=True` para que el modelo ajustara las probabilidades a priori según la frecuencia real de cada clase, decisión que resultó más efectiva que los **balanceos artificiales**, los cuales deterioraron significativamente el rendimiento. En conjunto, los resultados confirman que Naive Bayes responde bien ante clases frecuentes y léxicamente definidas, aunque requiere enfoques más avanzados más allá que un simple balanceo pero no siendo la mejor respuesta como algoritmo.

4.6.2. LinearSVC (C = 1.0, loss = “squared_hinge”, dual = False, max_iter = 2000, random_state = 42)

Tabla 11

Reporte de clasificación en validación - LinearSVC

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.5576	0.8037	0.6584	5146
DENUNCIA	0.6423	1.0000	0.7822	361
OTRO	0.5503	0.7081	0.6193	6585
QUEJA	0.5648	0.9867	0.7184	2174
RECLAMO	0.9425	0.7876	0.8581	37867
SUGERENCIA	0.4499	1.0000	0.6206	193

Tiempo de Entrenamiento: 649.16 segundos.

LinearSVC evidenció un rendimiento sólido, mostrando una mejora considerable en la identificación de categorías minoritarias. La clase **“RECLAMO”** obtuvo el mejor desempeño general (F1 = 0.8581), confirmando que el modelo discrimina eficazmente los textos de esta categoría. Por otro lado, **“DENUNCIA”** (F1 = 0.7822) y **“QUEJA”** (F1 = 0.7184) alcanzaron resultados altos en *recall* (1.0000 y 0.9867, respectivamente), lo que refleja una excelente capacidad para reconocer instancias verdaderas, aunque con cierta sobreasignación en clases menos representadas. En general, las categorías **“CONSULTA”**, **“OTRO”** y **“SUGERENCIA”** también mostraron F1-scores superiores a 0.60, lo que sugiere un comportamiento equilibrado del clasificador. Estos resultados indican que el **LinearSVC**, logra una **gran generalización**, manteniendo un desempeño robusto tanto en clases frecuentes como en las de menor soporte.

4.6.3. Random Forest (n_estimators = 100, min_samples_leaf = 10, max_features = 'sqrt', random_state = 42)

Tabla 12

Reporte de clasificación en validación - Random Forest

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.4311	0.1871	0.2610	5146
DENUNCIA	0.1549	0.1219	0.1364	361
OTRO	0.4260	0.0957	0.1562	6585
QUEJA	0.5217	0.2213	0.3107	2174
RECLAMO	0.7596	0.9496	0.8440	37867
SUGERENCIA	0.0580	0.0207	0.0305	193

Tiempo de Entrenamiento: 779.78 segundos.

Los resultados obtenidos con el algoritmo *Random Forest*, tras aplicar la técnica de balanceo artificial *SMOTE*, evidencian una mejora limitada en el desempeño general del modelo. Aunque la categoría *RECLAMO* mantiene un rendimiento elevado con un *f1-score* de 0.8440 gracias a su alta representatividad en el conjunto de datos, las demás clases presentan valores considerablemente bajos, especialmente *SUGERENCIA* (0.0305) y *DENUNCIA* (0.1364), lo que sugiere una dificultad del modelo para generalizar en categorías minoritarias. A pesar del intento de compensar el desbalance mediante *SMOTE*, el modelo tiende a sobreajustarse a la clase mayoritaria, indicando que el balanceo sintético no fue suficiente para mejorar la discriminación multiclase en este escenario.

4.6.4. Regresión Logística (solver = "lbfgs", penalty = "l2", C = 7, max_iter = 1000, random_state = 42)

Tabla 13

Reporte de clasificación en validación - Logistic Regression

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.5848	0.7686	0.6642	5146
DENUNCIA	0.6139	1.0000	0.7608	361
OTRO	0.5653	0.6867	0.6201	6585
QUEJA	0.5867	0.9696	0.7311	2174
RECLAMO	0.9288	0.8074	0.8639	37867
SUGERENCIA	0.4159	1.0000	0.5875	193

Tiempo de Entrenamiento: 890.18 segundos.

En el caso de la *Regresión Logística*, los resultados muestran un rendimiento sólido y equilibrado entre precisión y exhaustividad, con un desempeño especialmente notable en la clase *RECLAMO* (f1-score de 0.8639), seguida de *QUEJA* (0.7311) y *DENUNCIA* (0.7608). Las categorías minoritarias, como *SUGERENCIA* y *CONSULTA*, también alcanzan valores aceptables de f1-score (0.5875 y 0.6642 respectivamente), lo que evidencia una mejor capacidad del modelo para distinguir entre clases que en algoritmos previos. En conjunto, la *Regresión Logística* logra un equilibrio entre generalización y sensibilidad hacia las clases menos representadas, constituyéndose como una de las alternativas más consistentes dentro del conjunto de pruebas realizadas.

4.6.5. Agente de Claude AI

Tabla 14

Reporte de clasificación en validación - Claude AI

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.5712	0.6289	0.5986	984
DENUNCIA	0.5087	0.5725	0.5387	69
OTRO	0.4623	0.5178	0.4885	1,259
QUEJA	0.6014	0.6627	0.6306	415
RECLAMO	0.7834	0.7412	0.7617	7,238
SUGERENCIA	0.4791	0.5405	0.5081	37

Tiempo de Clasificación: 242.37 segundos.

El modelo LLM (Claude) mostró un desempeño heterogéneo en la clasificación de textos. La categoría “RECLAMO” alcanzó los mejores resultados (precisión = 0.7834, recall = 0.7412, F1 = 0.7617), mientras que “OTRO” presentó los valores más bajos (precisión = 0.4623, recall = 0.5178, F1 = 0.4885). Las categorías minoritarias “DENUNCIA” y “SUGERENCIA” obtuvieron F1-scores moderados de 0.5387 y 0.5081 respectivamente. La categoría “QUEJA” alcanzó valores de precisión = 0.6014, recall = 0.6627 y F1 = 0.6306, mientras que “CONSULTA” obtuvo precisión = 0.5712, recall = 0.6289 y F1 = 0.5986. En conjunto, el modelo evidencia variabilidad en su capacidad de clasificación entre categorías, con un desempeño considerable.

4.7. Simulación de Modelo de Clasificación Manual

La Tabla [15](#) presenta los resultados agregados de la simulación respecto al tiempo de clasificación manual.

En la Tabla [16](#) se detallan los resultados desagregados por tipo de mensaje.

Tabla 15*Estadísticas del tiempo de clasificación manual (N = 50,000 casos)*

Métrica	Valor
Tiempo mínimo general	2 días
Tiempo promedio	18.3 días
Tiempo máximo general	32 días
Desviación estándar	9.08 días
% clasificados en ≤ 7 días	16.8 %
% clasificados en ≤ 10 días	31.2 %
% demorados (> 20 días)	41.3 %

Tabla 16*Resultados por tipo de mensaje - Tiempo de clasificación*

Tipo	Casos	Proporción (%)	Promedio (días)	% > 20 días
RECLAMO	37,867	75.7	18.4	40.9
CONSULTA	5,146	10.3	17.8	39.5
QUEJA	2,174	4.4	19.1	44.7
OTRO	6,585	13.2	17.6	38.2
DENUNCIA	361	0.7	19.5	46.8
SUGERENCIA	193	0.4	16.9	35.2

Los resultados evidencian que el proceso de clasificación manual presenta importantes ineficiencias. El tiempo promedio para clasificar un mensaje es de 18.3 días hábiles (DE = 9.08), con un 41.3 % de los casos que tardan más de 20 días solo en ser clasificados. Los reclamos representan la categoría predominante (75.7 % del total), mientras que las denuncias presentan los mayores tiempos de clasificación (M = 19.5 días, 46.8 % > 20 días). La alta variabilidad en la revisión inicial (rango: 1-30 días) genera cuellos de botella críticos, donde solo el 16.8 % de los casos son clasificados dentro de la primera semana.

Estas cifras reflejan las graves demoras que ocurren en la etapa inicial del proceso, afectando significativamente la percepción ciudadana sobre la eficiencia del servicio incluso antes de iniciar la atención sustantiva de los casos.

Capítulo V

Resultados y Discusión

En este capítulo se presentan y analizan los resultados derivados de la experimentación con los distintos modelos de *machine learning* descritos en la fase anterior. Los resultados se organizan de manera que permitan evidenciar la mejora en la clasificación automática de reclamos, la eficiencia operativa del sistema y la validación de las hipótesis planteadas. Además, se incluyen comparaciones cruzadas y por modelo para visualizar las diferencias en desempeño según el tipo de reclamo.

5.1. Resultados de la Clasificación

5.1.1. Comparación Cruzada por Categoría

Para comprender de forma más integral el comportamiento de los modelos, se construyeron tablas que muestran cómo cada modelo respondió dentro de la categoría más presente y la de menor presencia.

Comparación en la Categoría “Reclamo”

Tabla 17

Comparación de los modelos en la categoría Reclamo

Modelo	Precisión	Recall	F1-Score	Soporte
Naive Bayes	0.8119	0.8737	0.8417	37867
Regresión Logística	0.9288	0.8074	0.8639	37867
Linear SVC	0.9425	0.7876	0.8581	37867
Random Forest	0.7596	0.9496	0.8440	37867
Claude AI	0.7834	0.7412	0.7617	7238

La categoría *Reclamo* concentra la mayor cantidad de registros, logrando resultados elevados en la mayoría de modelos. Los modelos tradicionales optimizados (*Regresión Logística*, *Linear SVC*) alcanzaron F1-scores superiores a 0.85, mientras que *Claude AI* obtuvo 0.7617, evidenciando limitaciones sin entrenamiento específico en el dominio.

Comparación en la Categoría “Sugerencia”

Tabla 18

Comparación de los modelos en la categoría Sugerencia

Modelo	Precisión	Recall	F1-score	
Naive Bayes	0.0753	0.0363	0.0490	193
Regresión Logística	0.4159	1.0000	0.5875	193
Linear SVC	0.4499	1.0000	0.6206	193
Random Forest	0.0580	0.0207	0.0305	193
Claude AI	0.4791	0.5405	0.5081	37

En la categoría *Sugerencia*, *Linear SVC* y *Regresión Logística* alcanzaron los mejores F1-scores (0.6206 y 0.5875 respectivamente), mientras que *Claude AI* obtuvo resultados intermedios (F1 = 0.5081), quedando por debajo de los modelos optimizados y evidenciando las limitaciones de operar sin entrenamiento específico en el dominio.

El patrón general evidencia la importancia del entrenamiento supervisado con datos específicos. Los modelos optimizados (*Linear SVC*, *Regresión Logística*) muestran resultados consistentemente superiores en todas las categorías, mientras que *Claude AI* presenta limitaciones significativas sin ajuste al dominio, especialmente en categorías ambiguas o minoritarias acompañado a su poca recepción de datos.

5.1.2. Mejor Modelo

Tabla 19

Desempeño del modelo Regresión Logística para clasificar categorías

Clase	Precisión	Recall	F1-Score	Soporte
CONSULTA	0.5848	0.7686	0.6642	5146
DENUNCIA	0.6139	1.0000	0.7608	361
OTRO	0.5653	0.6867	0.6201	6585
QUEJA	0.5867	0.9696	0.7311	2174
RECLAMO	0.9288	0.8074	0.8639	37867
SUGERENCIA	0.4159	1.0000	0.5875	193

El modelo de Regresión Logística logró una mejora sustancial, especialmente en la categoría *Consulta*. Su comportamiento más balanceado se debe al uso del esquema

TF-IDF, que permite asignar mayor peso a términos relevantes, estabilizando los límites de decisión. Las categorías con lenguaje más formal o técnico fueron clasificadas con mayor acierto.

5.1.3. Tiempos de entrenamiento y consideración computacional

La selección de un modelo de clasificación para despliegue productivo requiere equilibrar dos dimensiones críticas: (1) el desempeño predictivo, medido en términos de precisión y capacidad de generalización, y (2) los costos computacionales, expresados principalmente en tiempo de entrenamiento, recursos de hardware e inferencia.

Durante la experimentación, se registraron los tiempos de entrenamiento de los principales modelos, expresados en segundos, los cuales se muestran en la Tabla 20. Para el caso de *Claude AI (LLM)*, al no requerir entrenamiento, se reporta el tiempo de clasificación del conjunto de validación.

Tabla 20

Tiempos de entrenamiento observados (experimentales)

Modelo	Tiempo de entrenamiento (s)
Naive Bayes	0.97
Linear SVC	649.16
Random Forest	779.78
Regresión Logística	890.18
Claude AI (LLM)*	242.37

*Tiempo de clasificación (sin entrenamiento)

El modelo *Naive Bayes* destaca por su extrema rapidez, debido a su naturaleza probabilística y la ausencia de procesos iterativos de optimización. En contraste, la *Regresión Logística* y el *Linear SVC* muestran mayores tiempos de cómputo, asociados a la convergencia de sus funciones de costo mediante métodos graduales.

Un aspecto relevante es que, incluso considerando el modelo más exigente (Regresión Logística, con 890.18 s $\approx$ 14.8 minutos), el tiempo total de entrenamiento sigue siendo marginal frente a los plazos de atención observados en el proceso manual tradicional. Según el análisis del proceso manual, la clasificación de las 52,326 reclamaciones son de 18.3 días hábiles promedio.

En síntesis, los resultados demuestran que los modelos tradicionales optimizados —en especial *Linear SVC* y *Regresión Logística*— ofrecen la mejor relación entre tiempo de cómputo, desempeño predictivo y escalabilidad operativa. Estos modelos permiten reemplazar procesos manuales que tomarían trabajo acumulado por rutinas automati-

zadas capaces de procesar decenas de miles de casos en minutos, con un costo temporal y de recursos prácticamente despreciable.

5.2. Impacto en la Gestión Pública

Los beneficios observados en esta investigación trascienden el ámbito técnico y se extienden hacia la mejora operativa sustancial de los servicios públicos, como se detalla a continuación:

Reducción de tiempos de atención

La implementación del sistema automatizado reduce significativamente los tiempos de procesamiento inicial, de un promedio de 18.3 días hábiles en el proceso manual a minutos por lote. Esta reducción habilita procesos administrativos más ágiles y enfocados en la resolución sustantiva de los casos.

Consistencia en la clasificación

Al eliminar la interpretación subjetiva del funcionario mediante algoritmos estandarizados, se garantiza mayor trazabilidad, uniformidad y equidad en el tratamiento de las comunicaciones ciudadanas, reduciendo la variabilidad en los criterios de clasificación.

Escalabilidad del sistema

Los modelos evaluados demuestran capacidad de adaptación eficiente al incremento en el volumen de registros, permitiendo manejar cargas de trabajo crecientes sin requerir ampliaciones proporcionales en personal operativo.

Optimización de recursos institucionales

La disminución del tiempo dedicado a labores de revisión y clasificación manual libera recursos humanos especializados para actividades de mayor valor agregado, focalizando los esfuerzos en la resolución efectiva de casos y mejorando sustancialmente la percepción ciudadana sobre la calidad de atención estatal.

5.3. Discusión de Resultados

La comparación global revela patrones diferenciados según el tipo de modelo empleado. Los algoritmos basados en supuestos estadísticos (*Naive Bayes*) mostraron una fuerte dependencia de la frecuencia léxica y desempeño limitado en categorías minoritarias, mientras que los modelos discriminativos (*Regresión Logística*, *Linear SVC*)

lograron capturar relaciones entre términos con mejor balance entre precisión y generalización, alcanzando los mejores resultados globales.

Por otro lado, los modelos de ensamblaje (*Random Forest*) introducen estabilidad mediante la construcción de múltiples árboles, pero su comprensión limitada del contexto lingüístico restringe su aplicabilidad en reclamos redactados de manera emocional o narrativa. Este fenómeno se observa en las categorías *Queja* y *Denuncia*, donde los árboles tienden a confundirse ante vocablos de semántica ambigua o de tono subjetivo.

El modelo *Claude AI (LLM)* presenta un comportamiento intermedio. A diferencia de los modelos probabilísticos básicos, incorpora capacidades de comprensión contextual del lenguaje natural, lo que le permite obtener resultados superiores a *Naive Bayes* y *Random Forest*. Sin embargo, la ausencia de entrenamiento específico con los datos del dominio limita significativamente su desempeño frente a modelos optimizados como *Linear SVC* y *Regresión Logística*, evidenciando que la disponibilidad de datos etiquetados y ajuste de hiperparámetros resulta fundamental para maximizar el rendimiento en tareas de clasificación supervisada.

5.4. Validación de Hipótesis

Los resultados obtenidos confirman que el desempeño de los modelos depende críticamente del entrenamiento con datos específicos del dominio y la optimización de hiperparámetros. Los modelos tradicionales optimizados (*Linear SVC*, *Regresión Logística*) superaron consistentemente a alternativas sin ajuste específico, validando la importancia del aprendizaje supervisado con datos etiquetados para maximizar precisión y coherencia contextual en tareas de clasificación.

- **Hipótesis Específica 1:** El diseño y entrenamiento de modelos de *machine learning* permite visualizar mejora y eficiencia en la clasificación de reclamos en un libro de reclamaciones virtual, asegurando precisión y relevancia en las categorías utilizadas. Los resultados experimentales confirman plenamente esta hipótesis. Los modelos entrenados específicamente con los datos del dominio (*Regresión Logística*, *Linear SVC*) alcanzaron F1-scores superiores a 0.78, demostrando mejoras sustanciales en estabilidad, precisión y coherencia interclase. La optimización de hiperparámetros ($C=7$ para *Regresión Logística*, $C=1.0$ para *Linear SVC*) y el uso de vectorización TF-IDF aumentaron significativamente la capacidad de clasificación. En contraste, modelos sin optimización específica (*Claude AI*, $F1=0.71$) evidenciaron limitaciones, validando que el entrenamiento supervisado adecuado es fundamental para el éxito del sistema.

- **Hipótesis Específica 2:** La implementación de técnicas de procesamiento de lenguaje natural (NLP) en un sistema de *machine learning* mejora la comprensión y categorización automática de los textos de reclamos. Los resultados validan esta hipótesis con matices importantes. Las técnicas de NLP (vectorización TF-IDF, n-gramas) aplicadas sobre modelos supervisados demostraron excelente efectividad, permitiendo a *Linear SVC* y *Regresión Logística* capturar patrones léxicos y contextuales con alta precisión. Sin embargo, la mera incorporación de capacidades de lenguaje natural sin entrenamiento específico (*Claude AI*) resultó insuficiente para superar modelos tradicionales bien calibrados, evidenciando que las técnicas de NLP deben complementarse con optimización supervisada y ajuste al dominio para maximizar su efectividad en tareas específicas de clasificación.

Los hallazgos confirman que la combinación de técnicas apropiadas de NLP con entrenamiento supervisado y optimización de hiperparámetros no solo incrementa la precisión técnica, sino también la eficiencia operativa del sistema. Los modelos tradicionales optimizados demostraron ser la solución más efectiva para la gestión automatizada de reclamos, ofreciendo el mejor balance entre desempeño predictivo, velocidad de procesamiento y viabilidad operativa, reforzando el valor del aprendizaje automático supervisado como herramienta estratégica en sistemas de clasificación de texto.

Capítulo VI

Conclusiones y Recomendaciones

6.1. Conclusiones

6.1.1. Conclusión General

La implementación y optimización de técnicas de machine learning mejora la clasificación y la gestión de reclamos en una plataforma de libro de reclamaciones virtual.

El estudio demostró que los modelos supervisados tradicionales, especialmente LinearSVC y Regresión Logística, ofrecen un equilibrio óptimo entre precisión, eficiencia computacional y capacidad de generalización. La aplicación de técnicas de balanceo mediante SMOTE permitió mejorar el rendimiento en clases minoritarias, aunque persisten desafíos en categorías con muy baja representación.

La integración de técnicas de NLP con modelos supervisados optimizados constituye una solución viable y escalable para la automatización de la gestión de reclamos en el sector público, reduciendo tiempos de atención de meses a minutos y mejorando significativamente la eficiencia operativa.

6.1.2. Conclusiones Específicas

1. Sobre el diseño y entrenamiento de modelos de machine learning

El diseño, entrenamiento y ajuste de modelos supervisados demostró mejoras significativas en la clasificación automática de reclamos en el libro de reclamaciones virtual. Se evaluaron algoritmos clásicos como Naive Bayes, Random Forest, Regresión Logística y LinearSVC, junto con el modelo de lenguaje Claude AI.

Entre los modelos tradicionales, la Regresión Logística mostró un rendimiento equilibrado, destacando en la clase reclamo con un F1-Score de 0.8639 y en denuncia con 0.7608. El LinearSVC logró un F1-Score de 0.8581 en reclamo y mostró una alta capacidad de generalización, con recall de 1.0 en clases como denuncia y sugerencia.

El modelo Claude AI (LLM) mostró un desempeño competitivo, especialmente en la categoría reclamo (F1-Score 0.7617), aunque su rendimiento fue inferior al de los modelos supervisados optimizados.

La aplicación de SMOTE para balanceo de datos permitió mejorar la detección de clases minoritarias, aunque se evidenció la necesidad de técnicas complementarias para categorías con soporte muy reducido.

Estos resultados validan la primera hipótesis específica, confirmando que el entrenamiento supervisado con datos específicos del dominio mejora la eficiencia y precisión en la clasificación de reclamos.

2. Sobre la implementación de técnicas de procesamiento de lenguaje natural (NLP)

La aplicación de técnicas de NLP —como limpieza textual, lematización, vectorización TF-IDF y generación de sentence embeddings— permitió una representación numérica adecuada de los textos, facilitando la clasificación automática.

Los modelos que utilizaron representaciones TF-IDF (LinearSVC, Regresión Logística, Naive Bayes) mostraron un mejor desempeño en comparación con aquellos que utilizaron embeddings contextuales sin ajuste específico, como en el caso de Random Forest.

La combinación de técnicas de NLP con modelos supervisados optimizados permitió capturar patrones léxicos y semánticos en los textos, mejorando la comprensión automática y la coherencia en la categorización multiclase.

Sin embargo, el uso de modelos de lenguaje como Claude AI sin ajuste específico mostró limitaciones en la clasificación de categorías ambiguas o minoritarias, evidenciando que la mera incorporación de capacidades de NLP no es suficiente sin un entrenamiento supervisado adecuado.

De este modo, se valida la segunda hipótesis específica, confirmando que la aplicación de técnicas de NLP mejora la categorización automática, siempre que se complemente con modelos entrenados específicamente para el dominio.

6.2. Recomendaciones

- Ampliar el conjunto de datos de entrenamiento con reclamos de diversas instituciones públicas y períodos temporales, para mejorar la generalización de los modelos.
- Explorar otras técnicas de balanceo de datos, como ADASYN o ensembles basados en muestreo, para mejorar el rendimiento en clases con muy bajo soporte como sugerencia.
- Implementar procesos de reentrenamiento periódico de los modelos con nuevos datos, asegurando su adaptación a cambios en el lenguaje y la aparición de nuevos patrones de reclamos.
- Probar agentes de lenguaje con mayor capacidad de contexto y volumen de datos, para evaluar su potencial en la clasificación de reclamos institucionales.
- Diseñar una interfaz de interoperabilidad (API) que permita integrar el clasificador automatizado con los sistemas institucionales de gestión de reclamos, garantizando trazabilidad y escalabilidad.
- Desarrollar programas de capacitación para el personal técnico y administrativo en el uso, evaluación y mantenimiento de sistemas basados en machine learning y NLP, promoviendo una adopción responsable y sostenible en el marco de la transformación digital del sector público.

Bibliografía

- Ain, Q., Shaukat, A., & Saif, U. (2022). NLP based Model for Classification of Complaints: Autonomous and Intelligent System. *ResearchGate*. <https://doi.org/10.1109/ICoDT255437.2022.9787456>
- Autoridad Nacional del Servicio Civil (SERVIR), Escuela Nacional de Administración Pública (ENAP) & Secretaría de Gestión Pública - PCM. (2022). Gestión de Reclamos en las entidades de la administración pública. <https://www.gob.pe/institucion/servir/informes-publicaciones/2613283-gestion-de-reclamos-en-las-entidades-de-la-administracion-publica>
- Barbierato, E., & Gatti, A. (2024). The Challenges of Machine Learning: A Critical Review. *Electronics*, 13(2), 416. <https://doi.org/10.3390/electronics13020416>
- Bhartendoo Vimal & RV College of Engineering. (2020). Application of Logistic Regression in Natural Language Processing. *International Journal of Engineering Research and*, V9(06), IJERTV9IS060095. <https://doi.org/10.17577/IJERTV9IS060095>
- Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python [ISBN: 9780596516499]. <https://www.oreilly.com/library/view/natural-language-processing/9780596803346/>
- Blandón Andrade, J. C., Castaño Toro, A., Morales Ríos, A., & Orozco Ospina, D. (2025). Computational Methods for Information Processing from Natural Language Complaint Processes—A Systematic Review [Number: 1 Publisher: Multidisciplinary Digital Publishing Institute]. *Computers*, 14(1), 28. <https://doi.org/10.3390/computers14010028>
- Bozyiğit, F., Dogan, O., & Kilinç, D. (2022). Categorization of Customer Complaints in Food Industry Using Machine Learning Approaches. *ResearchGate*. <https://doi.org/10.38016/jista.954098>
- De Jager, S. (2023). Semantic Noise and Conceptual Stagnation in Natural Language Processing. *Angelaki*, 28(3), 111-132. <https://doi.org/10.1080/0969725X.2023.2216555>

- Deza, A., & Atamtürk, A. (2022). Safe Screening for Logistic Regression with – Regularization: *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 119-126. <https://doi.org/10.5220/0011578100003335>
- Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, 113(7), 4903-4923. <https://doi.org/10.1007/s10994-022-06296-4>
- Flores Poma, J. L. (2023). Clasificación de quejas de los clientes empleando procesamiento de lenguaje natural: revisión sistemática de la literatura. *ResearchGate*. <https://doi.org/10.15381/rpcs.v5i2.27134>
- Gao, P., Sun, N., Wang, X., Yang, C., & Zitakis, R. (2023, agosto). NLP-based detection of systematic anomalies among the narratives of consumer complaints. <https://doi.org/10.48550/arXiv.2308.11138>
- Garcia Amador, L. O., Chacón González, J. E., & Molina Castro, R. D. (2022). Modelo de clasificación de peticiones, quejas, reclamos y sugerencias en el sector financiero [Publisher: Pontificia Universidad Javeriana]. <https://doi.org/10.11144/Javeriana.10554.61721>
- Giannopoulou, E., & Mitrou, N. (2020). An AI-Based Methodology for the Automatic Classification of a Multiclass Ebook Collection Using Information From the Tables of Contents. *IEEE Access*, 8, 218658-218675. <https://doi.org/10.1109/ACCESS.2020.3041651>
- Houssein, E., Mohamed, R., & Ali, A. (2024). Machine Learning Techniques for Biomedical Natural Language Processing: A Comprehensive Review. *ResearchGate*. <https://doi.org/10.1109/ACCESS.2021.3119621>
- IBM. (2021, septiembre). ¿Qué es el aprendizaje automático (ML)? | IBM. <https://www.ibm.com/mx-es/think/topics/machine-learning>
- Jalal, N., Mehmood, A., Choi, G. S., & Ashraf, I. (2022). A novel improved random forest for text classification using feature ranking and optimal number of trees. *Journal of King Saud University - Computer and Information Sciences*, 34(6), 2733-2742. <https://doi.org/10.1016/j.jksuci.2022.03.012>
- Kowalski, R., Esteve, M., & Jankin Mikhaylov, S. (2020). Improving public services by mining citizen feedback: An application of natural language processing. *Public Administration*, 98(4), 1011-1026. <https://doi.org/10.1111/padm.12656>
- Lopes-Cardoso, H., Osório, T. F., Barbosa, L. V., Rocha, G., Reis, L. P., Machado, J. P., & Oliveira, A. M. (2021). Robust Complaint Processing in Portuguese [Number: 12 Publisher: Multidisciplinary Digital Publishing Institute]. *Information*, 12(12), 525. <https://doi.org/10.3390/info12120525>

- Manning, C. D., Raghavan, P., & Schütze, H. (2008, julio). *Introduction to Information Retrieval* (1.^a ed.). Cambridge University Press. <https://doi.org/10.1017/CBO9780511809071>
- Mansoori, A., Tahat, K., Tahat, D. N., Habes, M., & Salloum, S. A. (2025). Optimizing News Categorization with Machine Learning: A Comprehensive Study Using Naive Bayes (MultinomialNB) Classifier [Series Title: Studies in Big Data]. En A. Hamdan (Ed.), *Achieving Sustainable Business through AI, Technology Education and Computer Science* (pp. 169-178, Vol. 158). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-70855-8_15
- Muñoz-Céspedes, E., Ibar-Alonso, R., & De Lorenzo Ros, S. (2021). Financial Literacy and Sustainable Consumer Behavior. *Sustainability*, 13(16), 9145. <https://doi.org/10.3390/su13169145>
- Naik, P. K., T, P., Chandru, S., Jaganath, S., & Balan, S. (2023). Consumer Complaints Classification Using Machine Learning & Deep Learning. *International Research Journal on Advanced Science Hub*, 5(05), 116-122. <https://doi.org/10.47392/irjash.2023.S015>
- Nirala, K. K., Singh, N. K., & Purani, V. S. (2022). A survey on providing customer and public administration based services using AI: chatbot. *Multimedia Tools and Applications*, 81(16), 22215-22246. <https://doi.org/10.1007/s11042-021-11458-y>
- Olusegun, R., Oladunni, T., Audu, H., Houkpati, Y., & Bengesi, S. (2023). Text Mining and Emotion Classification on Monkeypox Twitter Dataset: A Deep Learning-Natural Language Processing (NLP) Approach. *IEEE Access*, 11, 49882-49894. <https://doi.org/10.1109/ACCESS.2023.3277868>
- Özçift, A., Akarsu, K., Yumuk, F., & Söylemez, C. (2021). Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. *Automatika*, 62(2), 226-238. <https://doi.org/10.1080/00051144.2021.1922150>
- PCM. (2021). Norma técnica N.º 001-2021-PCM-SGP. <https://www.gob.pe/institucion/pcm/normas-legales/3113333-001-2021-pcm-sgp>
- Peixoto, L. H. R. (2021, diciembre). *Aprendizado de Máquina Aplicado no Atendimento de Reclamações de Clientes* [text]. Universidade de São Paulo. <https://doi.org/10.11606/D.55.2021.tde-04022022-172002>
- Pereira, D. S. M., Falcão, F., Nunes, A., Santos, N., Costa, P., & José Miguel Pêgo. (2023). Designing and building OSCEBot ® for virtual OSCE – Performance evaluation [Publisher: Taylor & Francis]. *Medical Education Online*, 28(1), 2228550. <https://doi.org/10.1080/10872981.2023.2228550>

- Qurashi, A. W., Farhat, Z. A., Holmes, V., & Johnson, A. P. (2023). New Avenues for Automated Railway Safety Information Processing in Enterprise Architecture: An NLP Approach. *IEEE Access*, 11, 44413-44424. <https://doi.org/10.1109/ACCESS.2023.3272610>
- Ramirez-Guillen, I. A. P., & Matzumura-Kasano, J. P. (2024). Manejo de reclamos en servicios de salud en un Hospital de Lima Norte. *Revista Peruana de Ciencias de la Salud*, 6(4), 263-267. <https://doi.org/10.37711/rpcs.2024.6.4.550>
- Sabiri, B., Khtira, A., El Asri, B., & Rhanoui, M. (2023). Analyzing BERT's Performance Compared to Traditional Text Classification Models: *Proceedings of the 25th International Conference on Enterprise Information Systems*, 572-582. <https://doi.org/10.5220/0011983100003467>
- Saha, A., Chaure, A., Aayush, Bodhe, A., & Bhagat, P. T. (2024). Automated Complaint Classification and Routing Using NLP and Machine Learning [Publisher: IJIRT]. *International Journal of Innovative Research in Technology*, 11(6), 756-759. <https://ijirt.org/Article?manuscript=169187>
- Sarkar, D. (2019). *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing*. Apress. <https://doi.org/10.1007/978-1-4842-4354-1>
- Schwaar, S., Diez, F., Trebing, M., & Witznick, N. (2025, abril). Study on Text Classification for Public Administration. <https://doi.org/10.48550/arXiv.2504.09111>
- Secretaría de Gestión Pública - PCM. (2024). Manual de Uso: Tablero de información del LdR Digital. <https://www.gob.pe/institucion/pcm/informes-publicaciones/6290499-manual-de-usotablero-de-informacion-del-ldr-digital>
- Søgaard, A. (2021). *Explainable Natural Language Processing*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-02180-0>
- Stock, J., Petersen, T., Behrendt, C.-A., Federrath, H., & Kreutzburg, T. (2022). Privatsphärefreundliches maschinelles Lernen [Publisher: Springer Berlin Heidelberg]. *Informatik Spektrum*, 45(2), 70-79. <https://doi.org/10.1007/s00287-022-01438-3>
- VanderPlas, J. (2022). Python Data Science Handbook [ISBN: 9781098121228]. <https://www.oreilly.com/library/view/python-data-science/9781098121211/>
- Wang, D., Su, J., & Yu, H. (2020). Feature Extraction and Analysis of Natural Language Processing for Deep Learning English Language. *IEEE Access*, 8, 46335-46345. <https://doi.org/10.1109/ACCESS.2020.2974101>

Anexos

Tabla 21

Recursos experimentales: dataset y código fuente para su uso libre

URL	Descripción
<i>Dataset</i>	
https://drive.google.com/drive/folders/1drST-jhKbJpWU0ePCBVwe7F-FTFZBUQ2?usp=drive_link	Dataset <i>Reclamos_DT</i> en la carpeta TESIS MAESTRIA DATA
<i>Código Fuente</i>	
https://colab.research.google.com/drive/1Hb5C4C1Q6_I4XBU9wPE2E2eaTKLi2GBy?usp=sharing	Código <i>Tesis_NLP_Thalia.ipynb</i> de las pruebas de los modelos de aprendizaje automático.

Fuente. Elaboración Propia.

Código Fuente

```
# Instalación de librerías
pip install openpyxl
pip install pandas

# Carga del dataset
from google.colab import files
uploaded = files.upload()
file_name = list(uploaded.keys())[0]
file_name = "Reclamos_DT.xlsx"

import pandas as pd
data = pd.read_excel(file_name)

# Observación previa del dataset
conteo = (
    data["Tipo_registro"]
    .value_counts()
    .reset_index()
    .rename(columns={"index": "Tipo de Registro", "
                    Tipo_registro": "Cantidad"})
)
print(conteo)

# Procesamiento de Lenguaje Natural - Limpieza del Dataset
data.shape
data.isnull().sum()
data = data.dropna()
data.shape

# Estandarización de Etiquetas
data = data.rename(columns={"Tipo_registro": "Registro"})

registros_unicos = (
    data["Registro"]
    .dropna()
    .str.extract(r"^([^\s:]+)")
    .dropna()
    [0].unique().tolist()
)
print(registros_unicos)
```

```

data["Registro"] = data["Registro"].fillna("").str.strip().str
.lower()
data["Registro"] = data["Registro"].apply(
    lambda x: "otro" if x.startswith("otro") or x.startswith("
        acceso") else x
)
print(data["Registro"].unique())

data["Registro"] = data["Registro"].str.upper()
print(data["Registro"].unique())

conteo = (
    data["Registro"]
    .value_counts()
    .reset_index()
    .rename(columns={"index": "Tipo de Registro", "Registro":
        "Cantidad"})
)
print(conteo)

# Tokenizaci n
!python -m spacy download es_core_news_md
import pandas as pd
import spacy
import string
import numpy as np

data['Description'] = data['Description'].astype(str)

def preprocessor_textos(textos):
    nlp = spacy.load("es_core_news_md")
    textos_limpios = []
    for doc in nlp.pipe(textos, batch_size=256, n_process=-1):
        tokens_lematizados = [
            token.lemma_.lower() for token in doc
            if not token.is_stop and not token.is_punct and
            not token.is_space and not token.like_num
        ]
        textos_limpios.append(" ".join(tokens_lematizados))
    return textos_limpios

```

```

# Vectorizaci3n TF-IDF
!pip install scikit-learn
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizador_tfidf = TfidfVectorizer(
    ngram_range=(1, 2),
    sublinear_tf=True,
    min_df=5,
    max_df=0.9,
    max_features=50000
)
X_final = vectorizador_tfidf.fit_transform(data['Description'
])
y_final = data['Registro']

# Vectorizaci3n Sentence Embedding con Transformers
!pip install sentence-transformers numpy
!pip install torch torchvision torchaudio --index-url https://
download.pytorch.org/whl/cu121

from sentence_transformers import SentenceTransformer
import numpy as np
import time, gc

TEXT_COLUMN = "Description"
MODEL_NAME = "paraphrase-multilingual-MiniLM-L12-v2"
BATCH_SIZE = 128
CHUNK_SIZE = 50000

texts = data[TEXT_COLUMN].astype(str).tolist()
model = SentenceTransformer(MODEL_NAME, device="cuda")

total = len(texts)
n_chunks = (total + CHUNK_SIZE - 1) // CHUNK_SIZE
print(f"Procesando {total} textos en {n_chunks} partes de {
    CHUNK_SIZE}...")

for i in range(n_chunks):
    start_idx = i * CHUNK_SIZE
    end_idx = min((i + 1) * CHUNK_SIZE, total)

```

```

subset = texts[start_idx:end_idx]

print(f"\nParte {i+1}/{n_chunks}: textos {start_idx}-{
    end_idx}")
start = time.time()
embs = model.encode(subset, batch_size=BATCH_SIZE,
    show_progress_bar=True)
end = time.time()

np.save(f"embeddings_part_{i+1}.np", embs)
print(f"Guardado embeddings_part_{i+1}.np ({embs.shape})
    - {(end - start)/60:.2f} min")

del embs
gc.collect()

# Carga y concatenaci n de embeddings
import numpy as np
import glob

embedding_files = sorted(glob.glob("embeddings_part_*.np"))
embeddings_list = []
for file in embedding_files:
    print(f"Cargando {file}...")
    embs = np.load(file)
    embeddings_list.append(embs)

X_all = np.vstack(embeddings_list)
print(f"Shape total de embeddings: {X_all.shape}")
y_all = data["Registro"].values

# Partici n de datos
from sklearn.model_selection import train_test_split

# Partici n de la Vectorizaci n TF-IDF
X_train, X_test, y_train, y_test = train_test_split(
    X_final,
    y_final,
    test_size=0.2,
    random_state=42,
    stratify=y_final

```

```

)

# Partici n de la Vectorizaci n Embeddings
X_trainr, X_testr, y_trainr, y_testr = train_test_split(
    X_all, y_all, test_size=0.2, random_state=42, stratify=
    y_all
)

# Balanceo de Datos
!pip install imbalanced-learn

from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='auto', random_state=42,
    k_neighbors=5)
X_bal, y_bal = smote.fit_resample(X_train, y_train)

smote = SMOTE(sampling_strategy='auto', random_state=42,
    k_neighbors=5)
X_balr, y_balr = smote.fit_resample(X_trainr, y_trainr)

# M tricas
import time
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report,
    ConfusionMatrixDisplay

# Naive Bayes
from sklearn.naive_bayes import MultinomialNB

print("\n--- 4. Entrenando Naive Bayes ---")
start_time = time.time()

clf_nb = MultinomialNB(alpha=0.0098, fit_prior=True)
clf_nb.fit(X_train, y_train)
end_time = time.time()
print(f"Entrenamiento completado en {end_time - start_time:.2f
    } segundos.")

print("\n-- Reporte de Validaci n: Naive Bayes --")
y_pred_nb = clf_nb.predict(X_test)

```

```

print(classification_report(y_test, y_pred_nb, digits=4))

fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_nb, ax=
    ax, cmap='Blues')
ax.set_title("Matriz de Confusi n - Naive Bayes")
plt.xticks(rotation=45)
plt.show()

# Linear SVC
from sklearn.svm import LinearSVC

print("\n--- 2. Entrenando Linear SVC ---")
start_time = time.time()

clf_svc = LinearSVC(
    C=1.0, loss='squared_hinge', dual=False, max_iter=2000,
    random_state=42
)
clf_svc.fit(X_bal, y_bal)
end_time = time.time()
print(f"Entrenamiento completado en {end_time - start_time:.2f
    } segundos.")

print("\n-- Reporte de Validaci n: Linear SVC --")
y_pred_svc = clf_svc.predict(X_test)
print(classification_report(y_test, y_pred_svc, digits=4))

fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_svc, ax
    =ax, cmap='Blues')
ax.set_title("Matriz de Confusi n - Linear SVC")
plt.xticks(rotation=45)
plt.show()

# Random Forest
import time
import gc
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,
    ConfusionMatrixDisplay

```

```

import matplotlib.pyplot as plt

print("\n--- 3. Entrenando Random Forest ---")
start_time = time.time()

clf_rf = RandomForestClassifier(
    n_estimators=100,
    min_samples_leaf=10,
    class_weight=None,
    max_features='sqrt',
    n_jobs=-1,
    random_state=42,
)

clf_rf.fit(X_balr, y_balr)
end_time = time.time()
print(f"Entrenamiento completado en {end_time - start_time:.2f} segundos.")

print("\n--- Reporte de Validaci n: Random Forest ---")
y_pred_rf = clf_rf.predict(X_testr)
print(classification_report(y_testr, y_pred_rf, digits=4))

fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_testr, y_pred_rf, ax=ax, cmap='Blues')
ax.set_title("Matriz de Confusi n - Random Forest")
plt.xticks(rotation=45)
plt.show()

del clf_rf
gc.collect()

# Regresi n Log stica
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, ConfusionMatrixDisplay
import time
import matplotlib.pyplot as plt

print("\n--- 4. Entrenando Regresi n Log stica ---")

```

```

start_time = time.time()

clf_lr = LogisticRegression(
    solver='lbfgs',
    class_weight=None,
    penalty='l2',
    C=7,
    max_iter=1000,
    n_jobs=-1,
    random_state=42
)

clf_lr.fit(X_bal, y_bal)
end_time = time.time()
print(f"Entrenamiento completado en {end_time - start_time:.2f}
      segundos.")

print("\n-- Reporte de Validaci3n: Regresi3n Log3stica --")
y_pred_logreg = clf_lr.predict(X_test)
print(classification_report(y_test, y_pred_logreg, digits=4))

fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_logreg,
    ax=ax, cmap='Purples')
ax.set_title("Matriz de Confusi3n - Regresi3n Log3stica")
plt.xticks(rotation=45)
plt.show()

```