Universidad Tecnológica de la Mixteca

Clasificación de la polaridad de reseñas de hoteles todo incluido mediante aprendizaje automático

Tesis

Para obtener el título de:
Ingeniero en Computación

Presenta:
Axel Isaac García González

Director:
Dr. Christian Eduardo Millán Hernández

Co-director:
Dr. Eduardo Sánchez Soto

Huajuapan de León, Oaxaca, junio de 2025

A mi madre y tía,

Ana y Carolina.

Por su amor incondicional y su apoyo constante

en cada paso de este camino.

Agradecimientos

Agradezco profundamente a mi familia por siempre apoyarme en todo, por estar presentes en cada paso de este camino y por motivarme a seguir adelante en mi carrera profesional.

A mis amigos, gracias por su compañía, consejos, las risas e increíbles momentos compartidos a lo largo de este proceso. A Margarita, con quien compartí tantas experiencias durante la carrera, siempre apoyándome y contribuyendo a que esta etapa fuera aún más significativa.

Expreso mi sincero agradecimiento al Dr. Christian Eduardo Millán Hernández, por su guía, su disposición constante y por todas esas conversaciones que no solo aportaron a este estudio, sino que también me ayudaron a encontrar mi vocación dentro del mundo de la Inteligencia Artificial.

También agradezco al Dr. Eduardo Sánchez Soto, por su tiempo y su apoyo constante durante el desarrollo de este estudio, así como por aquellas amenas clases en las que aprender se convertía en un disfrute y no en una carga.

A todos mis profesores, gracias por su paciencia, su compromiso y las enseñanzas que marcaron mi formación académica. También agradezco al MTCA. Moisés Emmanuel Ramírez Guzmán, por todo el apoyo brindado durante esta etapa, siempre dispuesto a ayudarme en cada momento.

Finalmente, a la Universidad Tecnológica de la Mixteca, por brindarme los recursos, el espacio y las oportunidades necesarias para hacer realidad esta meta.

Índice general

Índice de Figuras iv

Índice de Tablas v

Resumen vii

1 Introducción 1

1.1 Antecedentes . . . 1
1.2 Planteamiento del problema . . . 6
1.3 Justificación . . . 7
1.4 Hipótesis . . . 8
1.5 Objetivos . . . 8
1.5.1 Objetivo general . . . 8
1.5.2 Objetivos específicos . . . 9
1.6 Metas . . . 9
1.7 Estructura de la tesis . . . 10

2 Marco Teórico 11

2.1 Procesamiento de Lenguaje Natural . . . 11
2.1.1 Análisis de Sentimientos . . . 12
2.1.2 Extracción y selección de características en datos textuales . . . 14
2.2 Balanceo de clases . . . 19
2.2.1 Submuestreo . . . 21
2.2.2 Sobremuestreo . . . 22
2.2.3 Aumento de datos basado en manipulación textual . . . 26
2.3 Aprendizaje Automático . . . 28
2.3.1 Regresión Logística . . . 29
2.3.2 Clasificador Bayesiano Ingenuo . . . 30
2.3.3 Máquina de Soporte Vectorial . . . 31
2.4 Evaluación de rendimiento . . . 33
2.4.1 Métricas . . . 34
2.4.2 Validación cruzada . . . 38
2.5 Trabajos relacionados . . . 40
2.5.1 Evaluación mediante Métricas Combinadas . . . 40
2.5.2 Evaluación por Métricas de Rango . . . 41

2.6 Resumen 42

3 Método Propuesto **45**

3.1 Descripción General del Método 45
3.2 Etapas del Método Propuesto 48
3.3 Resumen 52

4 Resultados **55**

4.1 Entorno de desarrollo y experimentación 56
4.2 Reseñas de hoteles todo incluido 57
4.2.1 Construcción de la base de datos 57
4.2.2 Análisis exploratorio de la base de datos 61
4.3 Preprocesamiento de la base de datos 69
4.3.1 Separación de los datos 69
4.3.2 Limpieza de los datos textuales 70
4.3.3 Extracción de características 71
4.3.4 Balanceo de clases 73
4.3.5 Selección de características 74
4.4 Experimento 1: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado A 76
4.4.1 Entrenamiento de modelos de clasificación 77
4.4.2 Evaluación del rendimiento de los modelos 86
4.4.3 Análisis y comparación de resultados 92
4.5 Experimento 2: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado B 95
4.5.1 Entrenamiento de modelos de clasificación 95
4.5.2 Evaluación del rendimiento de los modelos 104
4.5.3 Análisis y comparación de resultados 110

5 Conclusiones **113**

5.1 Aportaciones 115
5.2 Trabajo a futuro 116

Bibliografía **117**

Índice de figuras

2.1 *El proceso genérico del análisis de sentimientos* . . . 13
2.2 *Niveles de análisis de sentimientos* . . . 14
2.3 *Representación del proceso para construir una bolsa de palabras* . . . 16
2.4 *Visualización de un conjunto de datos desbalanceado* . . . 20
2.5 *Visualización del resultado de aplicar RUS a un conjunto de datos desbalanceado* . . . 21
2.6 *Creación de una instancia sintética con SMOTE* . . . 23
2.7 *Visualización del resultado de aplicar SMOTE a un conjunto de datos desbalanceado* . . . 24
2.8 *Visualización del resultado de aplicar DEBOHID a un conjunto de datos desbalanceado* 26
2.9 *Ejemplo de etiquetado de instancias para detección de spam* . . . 29
2.10 *Representación de un modelo de SVM en un espacio de dos dimensiones* . . . 32
2.11 *Ejemplos de subajuste, buen ajuste y sobreajuste* . . . 34
2.12 *Representación de una matriz de confusión para clasificación binaria* . . . 35
2.13 *División de datos usando K-Fold con* $K = 5$ . . . 39

3.1 *Esquema del método propuesto para la investigación* . . . 46

4.1 *Ejemplo de una reseña para el hotel Dreams Huatulco Resort & Spa* . . . 58
4.2 *Ejemplo de una reseña extraída para el hotel Barceló Huatulco* . . . 59
4.3 *Ejemplo de reseñas duplicadas en los datos* . . . 60
4.4 *Distribución de puntuaciones en la base de datos HuatulcoResortReviews* . . . 62
4.5 *Distribución de polaridades para ambos etiquetados en HuatulcoResortReviews* . . . 63
4.6 *Frecuencias de ocurrencias de caracteres y palabras en Etiquetado A en HuatulcoResortReviews* . . . 64
4.7 *Frecuencias de ocurrencias de caracteres y palabras en Etiquetado B en HuatulcoResortReviews* . . . 65
4.8 *Nubes de n-gramas por polaridad en Etiquetado A en HuatulcoResortReviews* . . . 66
4.9 *Nubes de n-gramas por polaridad en Etiquetado B en HuatulcoResortReviews* . . . 67
4.10 *Representación en dos dimensiones de las reseñas en HuatulcoResortReviews mediante t-SNE* . . . 68
4.11 *Proceso de limpieza para los datos textuales* . . . 71
4.12 *Ejemplos del cálculo del factor IGM* . . . 73
4.13 *Resultados de la selección de las mejores k características en Etiquetado A* . . . 75
4.14 *Resultados de la selección de las mejores k características en Etiquetado B* . . . 76
4.15 *Agrupación de resultados por técnica de vectorización en Etiquetado A* . . . 84
4.16 *Agrupación de resultados por técnica de balanceo de clases en Etiquetado A* . . . 85

4.17 *Matriz de confusión del modelo TF-IGM/DEBOHID/NB para Etiquetado A* 88
4.18 *Matriz de confusión del modelo TF-IDF/EDA/SVM para Etiquetado A* 90
4.19 *Matriz de confusión del modelo TF-IDF/EDA/LR para Etiquetado A* 92
4.20 *Agrupación de resultados por técnica de vectorización en Etiquetado B* 102
4.21 *Agrupación de resultados por técnica de balanceo de clases en Etiquetado B* 103
4.22 *Matriz de confusión del modelo TF-IDF/RUS/NB para Etiquetado B* 106
4.23 *Matriz de confusión del modelo TF-IDF/EDA/SVM para Etiquetado B* 108
4.24 *Matriz de confusión del modelo TF-IDF/EDA/LR para Etiquetado B* 110

Índice de Tablas

2.1 *Ejemplos de textos generados aplicando EDA* 28

4.1 *Especificaciones técnicas del equipo de cómputo utilizado* 56
4.2 *Bibliotecas de Python utilizadas para la implementación* 57
4.3 *Conteo de caracteres problemáticos por atributo* 60
4.4 *Descripción de las columnas de la base de datos HuatulcoResortReviews* 61
4.5 *Distribución de clases en los conjuntos de entrenamiento y prueba para Etiquetado A* . . 69
4.6 *Distribución de clases en los conjuntos de entrenamiento y prueba para Etiquetado B* . . 70
4.7 *Validación cruzada de los modelos que utilizan TF-IDF y LR para Etiquetado A* 78
4.8 *Validación cruzada de los modelos que utilizan TF-IDF y NB para Etiquetado A* 79
4.9 *Validación cruzada de los modelos que utilizan TF-IDF y SVM para Etiquetado A* 80
4.10 *Validación cruzada de los modelos que utilizan TF-IGM y LR para Etiquetado A* 81
4.11 *Validación cruzada de los modelos que utilizan TF-IGM y NB para Etiquetado A* 82
4.12 *Validación cruzada de los modelos que utilizan TF-IGM y SVM para Etiquetado A* . . . 83
4.13 *Resultados de la evaluación del modelo TF-IGM/DEBOHID/NB para Etiquetado A* . . . 87
4.14 *Resultados de la evaluación del modelo TF-IDF/EDA/SVM para Etiquetado A* 89
4.15 *Resultados de la evaluación del modelo TF-IDF/EDA/LR para Etiquetado A* 91
4.16 *Comparación de resultados de mejores modelos y modelos baseline para Etiquetado A* . 93
4.17 *Validación cruzada de los modelos que utilizan TF-IDF y LR para Etiquetado B* 96
4.18 *Validación cruzada de los modelos que utilizan TF-IDF y NB para Etiquetado B* 97
4.19 *Validación cruzada de los modelos que utilizan TF-IDF y SVM para Etiquetado B* 98
4.20 *Validación cruzada de los modelos que utilizan TF-IGM y LR para Etiquetado B* 99
4.21 *Validación cruzada de los modelos que utilizan TF-IGM y NB para Etiquetado B* 100
4.22 *Validación cruzada de los modelos que utilizan TF-IGM y SVM para Etiquetado B* . . . 101
4.23 *Resultados de la evaluación del modelo TF-IDF/RUS/NB para Etiquetado B* 105
4.24 *Resultados de la evaluación del modelo TF-IDF/EDA/SVM para Etiquetado B* 107
4.25 *Resultados de la evaluación del modelo TF-IDF/EDA/LR para Etiquetado B* 109
4.26 *Comparación de resultados de mejores modelos y modelos baseline para Etiquetado B* . 111

Resumen

Esta tesis aborda la selección y evaluación de técnicas de Procesamiento de Lenguaje Natural y Aprendizaje Automático que permitan maximizar el rendimiento de modelos para la clasificación de polaridad en reseñas escritas en inglés, relacionadas con la experiencia de clientes en hoteles todo incluido ubicados en Bahías de Huatulco, Oaxaca. Una correcta identificación de la polaridad de las opiniones puede ser fundamental para la toma de decisiones orientadas a mejorar la calidad del servicio hotelero.

Se emplean distintos métodos de extracción de características, como TF-IDF y TF-IGM; técnicas de balanceo de clases, entre ellas Random Oversampling, Random Undersampling, SMOTE, DEBOHID y EDA; así como algoritmos de clasificación como Regresión Logística, el Clasificador Bayesiano Ingenuo y la Máquina de Soporte Vectorial. Además, se utiliza la métrica de Media Geométrica (G-Mean) para realizar una evaluación justa del rendimiento de los modelos generados. El método propuesto contempla la creación de una base de datos, su preprocesamiento, el entrenamiento de modelos y el análisis de los resultados obtenidos.

1 Introducción

Contenidos del Capítulo

1.1 Antecedentes 1
1.2 Planteamiento del problema 6
1.3 Justificación 7
1.4 Hipótesis 8
1.5 Objetivos 8
1.5.1 Objetivo general 8
1.5.2 Objetivos específicos 9
1.6 Metas 9
1.7 Estructura de la tesis 10

1.1 Antecedentes

El turismo es un fenómeno social y cultural que involucra el desplazamiento de personas de su ambiente cotidiano a otros lugares, por motivos personales, profesionales o de negocios[1]. En México, el Estado de Oaxaca es uno de los destinos turísticos más atractivos para visitantes nacionales e internacionales. En este Estado el turismo contribuye de manera sustancial a la generación de empleos en los principales destinos turísticos, como lo es Bahías de Huatulco, volviéndose una de las principales actividades que impulsa significativamente el desarrollo económico y social de la entidad [Congreso del Estado Libre y Soberano de Oaxaca 2021]. De acuerdo con cifras del año 2021, el PIB turístico supuso un aproximado de $25,751 millones de pesos, lo que representó el 10.32 % del PIB estatal preliminar para ese mismo año, que fue de $249,591 millones de pesos [Secretaría de Turismo del Estado de Oaxaca 2023].

Sin embargo, sin importar el lugar al que se desplacen, los turistas necesitan seleccio-

[1] WTO. (s.f.). Glosario de términos de turismo. Recuperado el 21 de agosto de 2024, de https://www.unwto.org/es/glosario-terminos-turisticos

nar el alojamiento o espacio donde residirán durante su estancia. Comúnmente, los sitios elegidos por los turistas para este fin son los hoteles. Entre la amplia variedad de hoteles disponibles, los todo incluido se caracterizan por ofrecer todo tipo de servicios como: habitación, comida, bares, entretenimiento, guardería, entre otras instalaciones especiales en el precio acordado a pagar [Márquez Reiter et al. 2023].

Por otra parte, los turistas, además de escoger el hotel donde se hospedarán, tienen que verificar que los servicios ofrecidos cumplan sus necesidades y expectativas, es decir, que sean de calidad [Maldonado y Hernández 2011]. En [Contreras Castañeda 2021; Tosun et al. 2015] se remarca la importancia de medir la calidad de los servicios en destinos turísticos como una ventaja competitiva para sobresalir entre la competencia. La presencia de calidad en los servicios genera una experiencia de compra positiva y aumenta la satisfacción de los clientes, lo que resulta en opiniones favorables que impulsan el crecimiento del proveedor del servicio. Por esta razón, es crucial para los hoteles diferenciar entre las opiniones de sus clientes, los turistas, a propósito de los servicios recibidos, para determinar si logran satisfacer sus necesidades y expectativas.

El auge y crecimiento del Internet ha traído consigo la creación de plataformas web, como Tripadvisor o Booking, en las que sus usuarios reseñan sus experiencias de viaje. Usualmente, las reseñas se expresan mediante puntos de vista, sentimientos y creencias del usuario, donde en adición se pone una calificación numérica que resume su opinión. Este tipo de dato textual suele ser referido como contenido generado por el usuario [Barreda y Bilgihan 2013] donde se utiliza una redacción de texto sin edición y que utiliza un lenguaje informal. Las plataformas utilizan estas opiniones como fuente de consulta y referencia con el fin de que los potenciales clientes tengan una opinión real de otros clientes y de esta manera puedan tomar la decisión sobre si el lugar es el ideal o no para hospedarse [Cherdouh et al. 2022; Shi y Li 2011].

En contraste, la tarea de procesar estos datos textuales, como parte de la mejora de los servicios por parte del proveedor, es una ardua labor, dada la cantidad de opiniones expresadas en texto. Automatizar la separación de las opiniones entre lo que se está haciendo bien, o, por el contrario, lo que no, representando esto último a las áreas donde la calidad del servicio decae, implica realizar un análisis para seleccionar aquellas que resulten de interés. Es decir, que las opiniones negativas pueden servir para la toma de decisiones basadas en

datos con el fin de formular soluciones relacionadas con la calidad de los servicios.

Realizar la tarea de forma manual implica leer individualmente cada reseña para extraer información útil, como su posible polaridad, es decir, si se trata de una opinión negativa, neutra o positiva. Tan solo para la plataforma Tripadvisor, de acuerdo con su reporte de transparencia de reseñas del año 2023, se tendría que analizar más de 30.2 millones de reseñas que han sido publicadas por sus miembros en ese año[2].

Una forma eficiente de abordar el problema de la clasificación de la polaridad de reseñas, con el fin de tomar decisiones para mejorar los servicios ofrecidos por los hoteles todo incluido, es mediante el uso del Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés). El NLP es un área de la inteligencia artificial, las ciencias de la computación y de la lingüística que actúa como puente entre los lenguajes naturales y las computadoras; permitiendo a estas últimas entender, procesar y analizar el lenguaje humano [Torfi et al. 2020]. El NLP provee las herramientas para realizar la clasificación de las reseñas, como por ejemplo, el análisis de sentimientos, un subcampo que se encarga de recuperar los sentimientos o polaridad expresada por un fragmento de texto a partir de su contenido [Wankhade et al. 2022].

En la literatura se han abordado distintas estrategias para realizar la clasificación de polaridad de conjuntos organizados de textos, o corpus, mediante el uso en conjunto del NLP con el aprendizaje automático. Estas estrategias varían principalmente en cuatro aspectos, que se describen a continuación.

El primer aspecto es la forma en que se hace la extracción de características del corpus para obtener representaciones numéricas de los datos textuales, como el uso de Bolsa de Palabras o Bag of Words (BoW) y sus variantes supervisadas y no supervisadas, como TF-IGM o TF-IDF, respectivamente [Chang et al. 2023; Chen et al. 2016; Dharma y Saragih 2022; Gazali Mahmud et al. 2023; Satriaji y Kusumaningrum 2018]; al igual que el uso de Embeddings tal como Word2Vec o Doc2Vec [Chang et al. 2023; Khamphakdee y Seresangtakul 2021].

El segundo aspecto se refiere a las técnicas de balanceo de datos, las cuales se utilizan

[2]Foley, B. (s.f.). 2023 Tripadvisor Review Transparency Report. Recuperado el 29 de agosto de 2024, de https://www.tripadvisor.com/TransparencyReport2023

para prevenir sesgos cuando hay clases mayoritarias, es decir, cuando el conjunto de datos está desbalanceado porque una clase cuenta con más ejemplos que otras. Este aspecto es un punto importante de destacar, ya que, en muchos casos, la distribución de los textos tiende a ser desproporcionada entre clases, inclinándose hacia uno de los extremos, ya sea al positivo o negativo. Algunas de las técnicas utilizadas para enfrentar el desbalance de datos en los trabajos previos son Synthetic Minority Oversampling Technique (SMOTE), Random Oversampling y Undersampling [Dharma y Saragih 2022; Gazali Mahmud et al. 2023; Satriaji y Kusumaningrum 2018]. Por otra parte, existen técnicas basadas en metaheurísticas como Differential Evolution Based Over-sampling for Highly Imbalanced Datasets (DEBOHID) e incluso se utiliza la generación de prototipos mediante algoritmos genéticos para crear conjuntos de datos reducidos que mantengan las proporciones de los datos reales [Hernández Martínez 2022, 18 de noviembre; Kaya et al. 2021].

El tercer aspecto a tomar en cuenta es el algoritmo de Aprendizaje Automático utilizado para realizar la clasificación. En los trabajos previos se han utilizado aquellos como k-Vecinos más Cercanos o kNN, por sus siglas en inglés [Gazali Mahmud et al. 2023], los Árboles de Decisión [Yordanova y Kabakchieva 2017], la Regresión Logística, el Clasificador Bayesiano Ingenuo o Naïve Bayes y la Máquina de Soporte Vectorial o SVM, por sus siglas en inglés [Dharma y Saragih 2022; Satriaji y Kusumaningrum 2018].

Como cuarto y último aspecto, encontramos a las métricas de evaluación del rendimiento de los modelos generados. Estas métricas son las que nos permiten tomar decisiones a partir de los resultados obtenidos al analizar la efectividad en la clasificación. Entre las métricas comunes vemos el uso de la Exactitud, Precisión, Sensibilidad y el F1, o F-Score [Dharma y Saragih 2022]. En cambio, cuando se presenta un desbalance de clases, los trabajos previos han optado por métricas que aseguren una evaluación justa y equilibrada, como la Media Geométrica o G-Mean. Además, también se ha recurrido al área bajo la curva ROC (ROC-AUC) como métrica complementaria [Gazali Mahmud et al. 2023; Satriaji y Kusumaningrum 2018].

En [Satriaji y Kusumaningrum 2018] realizaron un estudio sobre el impacto que tiene SMOTE, para realizar la clasificación de opiniones de hoteles. Hicieron uso de técnicas de extracción de características como Term Presence, Term Occurrence y TF-IDF. Como algoritmos de aprendizaje utilizaron la Regresión Logística, Naïve Bayes y SVM, evaluándolos

con la métrica G-Mean. Obtuvieron que el uso de Term Occurrence en conjunto con la Regresión Logística dio mejores resultados en promedio, alcanzando un puntaje de 81.65 % de G-Mean.

Para su investigación, [Dharma y Saragih 2022] compararon tres técnicas de extracción de características: BoW, TF-IDF e Improved TF-IDF, en la clasificación de reseñas de hoteles mediante análisis de sentimientos en tres clases, positiva, negativa y neutra. Como método de balanceo de clases, hicieron uso de un enfoque combinado de oversampling y undersampling. Utilizaron SVM como algoritmo de aprendizaje y evaluaron los resultados con las métricas de Exactitud, Precisión, Sensibilidad y F-Score. Como resultado, muestran que TF-IDF fue la técnica que produjo los mejores resultados, alcanzando una Exactitud de 71.75 %, una Precisión de 78.66 %, una Sensibilidad de 71.91 % y un F-Score de 70.08 %.

En [Gazali Mahmud et al. 2023] implementaron el algoritmo de kNN en combinación con SMOTE, con el objetivo de mejorar la precisión en la clasificación de reseñas de hoteles ubicados en Indonesia en positivas y negativas. Como técnicas de extracción de características hicieron uso de BoW, TF-IDF e Improved TF-IDF. El modelo fue entrenado con y sin balanceo de clases y evaluado con las métricas de Exactitud, Precisión, Sensibilidad y ROC-AUC. Sus resultados muestran que aplicar SMOTE para balancear las clases mejoró el rendimiento del modelo en comparación de no aplicarlo, alcanzando un valor de ROC-AUC que va de 82.1 % a 94.4 %, dependiendo de las reseñas del hotel utilizado para entrenar el modelo.

Tal como se ha descrito en los párrafos anteriores, los trabajos del estado del arte presentan diversas metodologías y técnicas diferentes para realizar la clasificación de reseñas. No obstante, la clasificación de la polaridad de reseñas en inglés sobre la experiencia de los clientes de servicios de hoteles todo incluido en Bahías de Huatulco, Oaxaca, requiere un enfoque adaptado a su contexto particular. Esta investigación tiene como finalidad proponer un método que permita crear modelos de aprendizaje automático que combinen múltiples técnicas de extracción de características y balanceo de datos, empleando métricas que permitan evaluar su desempeño de manera justa y equitativa.

1.2 Planteamiento del problema

El uso del aprendizaje automático como una manera de resolver problemas de predicción en la vida real, conlleva una serie de etapas para seleccionar aquellos modelos que resulten los más apropiados. Es importante considerar que no existe una metodología universal para resolver cualquier tipo de problema, ya que, dependiendo de las particularidades y el dominio del problema, distintos enfoques pueden mostrar un mejor rendimiento en comparación con otros.

En esta tesis se aborda el problema de seleccionar y evaluar aquellas técnicas que nos permitan maximizar el rendimiento de los modelos obtenidos mediante aprendizaje automático, para la clasificación de la polaridad de reseñas, con especial énfasis en la recuperación de las reseñas negativas sobre hoteles todo incluido. Dado que, se parte de la premisa de que en este tipo de textos se concentra la información útil para identificar áreas de oportunidad y mejora, se tiene como finalidad dar relevancia a la identificación de las reseñas negativas, sin descuidar la adecuada recuperación de las positivas. Estas servirán como una herramienta o guía para la toma de decisiones basadas en datos, orientadas a mejorar la calidad de los servicios ofrecidos por los hoteles todo incluido [Barreda y Bilgihan 2013].

Para sustentar este trabajo, en el estado del arte, se identifica que el enfoque supervisado es el más común para la tarea de clasificación de la polaridad de reseñas. En el estudio de [Satriaji y Kusumaningrum 2018] utilizaron la Regresión Logística, Naïve Bayes, y SVM, mientras que [Dharma y Saragih 2022] emplearon únicamente SVM como algoritmo central en su investigación. De manera similar a este último trabajo, en [Gazali Mahmud et al. 2023] hicieron uso de kNN como único algoritmo implementado.

El uso de distintos métodos y técnicas para maximizar el rendimiento en la clasificación, como las de extracción de características y balanceo de clases, son las principales similitudes entre los trabajos del estado del arte. No obstante, ninguno de estos trabajos detalla cómo seleccionar las técnicas más adecuadas ni aborda cuál de las métricas utilizadas es la más útil para evaluar el rendimiento de los modelos y, por consiguiente, tampoco el cómo elegir el mejor modelo para la tarea en cuestión. Por tal razón, se propone la siguiente

pregunta de investigación:

> ¿Qué técnicas de extracción de características y balanceo de clases, combinadas, pueden mejorar el desempeño de los algoritmos de aprendizaje automático en la clasificación de la polaridad positiva y negativa de reseñas en inglés sobre la experiencia de clientes en hoteles todo incluido?

1.3 Justificación

La Secretaría de Turismo del Estado de Oaxaca publicó que el PIB turístico representó el 10.32 % del PIB estatal preliminar del año 2021, lo que supuso un aproximado de $25,751 millones de pesos en ingresos [Secretaría de Turismo del Estado de Oaxaca 2023]. Dada la importancia del sector turístico para la economía del Estado, es importante analizar la satisfacción de los clientes con los servicios recibidos durante su estancia, con el fin de promover su mejora continua. La satisfacción o, por el contrario, la insatisfacción del cliente se relaciona con la calidad del servicio ofrecido por el hotel en cuestión.

Para mantener la lealtad de los clientes actuales y atraer a potenciales clientes nuevos, los hoteles deben observar y resolver a la brevedad cualquier problema presente en su servicio. Las reseñas de plataformas en línea son una de las principales fuentes de información para los hoteles, ya que estas reseñas son la retroalimentación dada por los clientes actuales, además de ser el primer acercamiento que tienen potenciales clientes a propósito del servicio, por lo que tienen un gran impacto en la imagen y la calidad del servicio del hotel [Mauri y Minazzi 2013].

Sin embargo, si se recurre a un análisis de datos manual, la gran cantidad de reseñas o datos textuales generados en plataformas como Tripadvisor o Booking, implica realizar un estudio complicado y costoso. No obstante, el uso del NLP en conjunto con el aprendizaje automático facilita el análisis automatizado de grandes volúmenes de datos, reduciendo costos en la clasificación de la polaridad de las reseñas a fin de extraer información útil para la toma de decisiones orientadas a aumentar la calidad del servicio [Chang et al. 2023].

Es necesario mencionar que el uso de estos métodos no busca reemplazar el trabajo de expertos en el área, sino proporcionar una herramienta para la toma de decisiones. De esta manera, el experto humano podrá proponer soluciones adecuadas a las necesidades particulares de los hoteles todo incluido de Bahías de Huatulco, Oaxaca.

Al tratarse Huatulco de una de las principales zonas turísticas del Estado de Oaxaca, los resultados de esta investigación pueden beneficiar directamente a la economía del sector turístico y a la población de la región. Dado que el turismo es una de las actividades económicas más importantes en Oaxaca, facilitar un análisis preciso de los datos textuales sobre las experiencias de los clientes permitirá a los administradores de hoteles o interesados tomar decisiones orientadas a mejorar sus servicios. Esto contribuirá a incrementar la satisfacción del turista y, en consecuencia, al fortalecimiento de la economía en Huatulco.

1.4 Hipótesis

El uso de una métrica como G-Mean, que equilibra la recuperación de las clases de polaridad positiva y negativa a comparación del F-Score, permitirá evaluar el impacto de las técnicas de extracción de características y balanceo de clases en modelos de aprendizaje automático para identificar la polaridad de reseñas en inglés sobre servicios de hoteles todo incluido.

1.5 Objetivos

1.5.1 Objetivo general

Clasificar la polaridad de reseñas en inglés sobre la experiencia de los clientes de servicios de hoteles todo incluido en Bahías de Huatulco, Oaxaca.

1.5.2 Objetivos específicos

1. Realizar una investigación documental de los métodos y técnicas utilizadas para la clasificación de polaridad de reseñas de servicios.
2. Crear un corpus de reseñas de hoteles todo incluido de Bahías de Huatulco, Oaxaca, en el idioma inglés, sobre la experiencia de sus clientes.
3. Proponer un método para explorar la implementación de distintas técnicas de procesamiento de datos textuales y de técnicas de balanceo de datos para crear modelos de aprendizaje automático para la clasificación de la polaridad tanto de reseñas negativas y positivas.
4. Evaluar los modelos obtenidos de los algoritmos de aprendizaje automático mediante las métricas apropiadas.
5. Realizar un análisis comparativo de los resultados obtenidos de la evaluación de los diferentes modelos.

1.6 Metas

1. Estudio de los métodos y técnicas de extracción de características, balanceo de datos, algoritmos de aprendizaje automático y métricas de evaluación utilizadas para la clasificación de textos.
2. Implementación de un método para crear modelos de aprendizaje automático para la clasificación de la polaridad de reseñas de hoteles todo incluido.
3. Construcción de un corpus de reseñas de hoteles todo incluido.
4. Redacción del documento de tesis para la obtención del título de Ingeniero en Computación.

1.7 Estructura de la tesis

En el capítulo 2 se presentan los fundamentos teóricos relevantes para el desarrollo de esta tesis, como el procesamiento de lenguaje natural y el aprendizaje automático, que involucra técnicas de extracción de características, de balanceo de datos, algoritmos de aprendizaje y métricas de evaluación. En el capítulo 3 se presenta el método propuesto para realizar la clasificación de la polaridad de las reseñas de hoteles todo incluido. El capítulo 4 presenta los resultados obtenidos a partir de la experimentación con el método propuesto, y el capítulo 5 expone las conclusiones derivadas de esta investigación.

2 Marco Teórico

Contenidos del Capítulo

2.1 Procesamiento de Lenguaje Natural . . . 11
2.1.1 Análisis de Sentimientos . . . 12
2.1.2 Extracción y selección de características en datos textuales . . . 14
2.2 Balanceo de clases . . . 19
2.2.1 Submuestreo . . . 21
2.2.2 Sobremuestreo . . . 22
2.2.3 Aumento de datos basado en manipulación textual . . . 26
2.3 Aprendizaje Automático . . . 28
2.3.1 Regresión Logística . . . 29
2.3.2 Clasificador Bayesiano Ingenuo . . . 30
2.3.3 Máquina de Soporte Vectorial . . . 31
2.4 Evaluación de rendimiento . . . 33
2.4.1 Métricas . . . 34
2.4.2 Validación cruzada . . . 38
2.5 Trabajos relacionados . . . 40
2.5.1 Evaluación mediante Métricas Combinadas . . . 40
2.5.2 Evaluación por Métricas de Rango . . . 41
2.6 Resumen . . . 42

En este capítulo se presentan los fundamentos teóricos necesarios para el desarrollo de la presente investigación.

2.1 Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés de *Natural Language Processing*) es un subcampo de las ciencias computacionales, la ingeniería y la inteligencia artificial, con raíces en la lingüística computacional. Su propósito principal es desarrollar

sistemas y aplicaciones que faciliten la interacción entre las máquinas o computadoras y los lenguajes desarrollados naturalmente por los seres humanos [Sarkar 2016].

Mediante el uso de técnicas de NLP, las computadoras adquieren la capacidad de interpretar y procesar el lenguaje humano, aún no de forma perfecta, lo que les permite generar respuestas o resultados adecuados en condiciones prácticas. Entre las aplicaciones más comunes se encuentran la traducción automática, los sistemas de reconocimiento de voz, la generación de resúmenes de texto y el análisis de texto [Torfi et al. 2020].

En particular, el análisis de texto se destaca debido a su capacidad para extraer patrones e información relevante de datos textuales. Debido a la naturaleza no estructurada del texto, es necesario transformarlo en representaciones numéricas que puedan ser procesadas por algoritmos de aprendizaje automático. Para ello, es necesario realizar una extracción de características que puedan ser representadas numéricamente de manera eficiente. Como paso previo, es fundamental aplicar un preprocesamiento al texto, que puede incluir tareas como la normalización, la eliminación de palabras vacías y la lematización, entre otras, con el objetivo de resaltar las características más relevantes del contenido [Sarkar 2016].

2.1.1 Análisis de Sentimientos

En el análisis de texto, el análisis de sentimientos (SA, por sus siglas en inglés de *Sentiment Analysis*) se enfoca en identificar y evaluar la polaridad emocional de los textos, clasificándolos en categorías. Se emplea principalmente para extraer información de textos subjetivos, como encuestas y opiniones, que reflejan emociones, actitudes y estados de ánimo humanos, determinando si la polaridad del texto es positiva, negativa o neutral [Sarkar 2016].

Las emociones son respuestas automáticas y no conscientes del cerebro ante estímulos que impactan la supervivencia, como amenazas o recompensas, y están mediadas por circuitos cerebrales como los de la amígdala. Por otro lado, los sentimientos son experiencias conscientes que surgen cuando la información sobre estas respuestas emocionales se integra en la memoria de trabajo junto con contextos cognitivos y recuerdos. La principal diferencia radica en que las emociones son procesos automáticos, mientras que los senti-

mientos son la interpretación consciente y subjetiva de esos procesos [LeDoux 2015].

El proceso genérico de SA se ilustra en la Figura **2.1**. Una vez que los datos han sido recopilados y extraídos de diversas fuentes y en diferentes formatos, se convierten en texto y se procesan mediante técnicas de NLP. Este procesamiento incluye etapas como el preprocesamiento del texto, la extracción de características y la selección de las mismas. La clasificación puede llevarse a cabo con enfoques como el aprendizaje automático; y, finalmente, la salida o los resultados se presentan mediante representaciones gráficas que faciliten su interpretación [Birjali et al. 2021]. Estos resultados pueden ser utilizados para tomar decisiones basadas en datos, análisis de tendencias o mejorar la experiencia del usuario en distintos contextos.

Figura 2.1
El proceso genérico del análisis de sentimientos

Nota. Fuente [Birjali et al. 2021].

El SA puede realizarse a nivel de aspecto, que busca identificar los sentimientos con respecto a aspectos específicos en entidades; a nivel de oración, o a nivel de documento, asignando puntajes a sentimientos positivos y negativos para etiquetar la polaridad final. En específico, el nivel de documento se realiza sobre un documento o texto completo, donde se le asigna una única polaridad [Wankhade et al. 2022]. En la Figura **2.2** observamos cada nivel mencionado, donde se va de lo más general, que es el nivel de documento, a lo más

específico, siendo este a nivel de aspecto.

Figura 2.2
Niveles de análisis de sentimientos

Nota. Fuente [Birjali et al. 2021].

2.1.2 Extracción y selección de características en datos textuales

En el caso de los datos textuales, la principal dificultad radica en cómo transformar el texto en representaciones numéricas que los algoritmos de aprendizaje automático puedan procesar y utilizar para realizar una predicción. La extracción de características es el proceso en el cual se identifican y extraen atributos medibles de los datos, siendo estos generalmente numéricos o categóricos. Las características permiten a los algoritmos aprender patrones de los datos.

El Modelo de Espacio Vectorial (VSM, por sus siglas en inglés de *Vector Space Model*) es un método para representar documentos textuales como vectores numéricos, donde cada dimensión corresponde a una palabra única y su peso refleja su importancia, como lo es su frecuencia de aparición [Sarkar 2016]. De manera formal, para un documento D en un espacio vectorial VS, su representación matemática es:

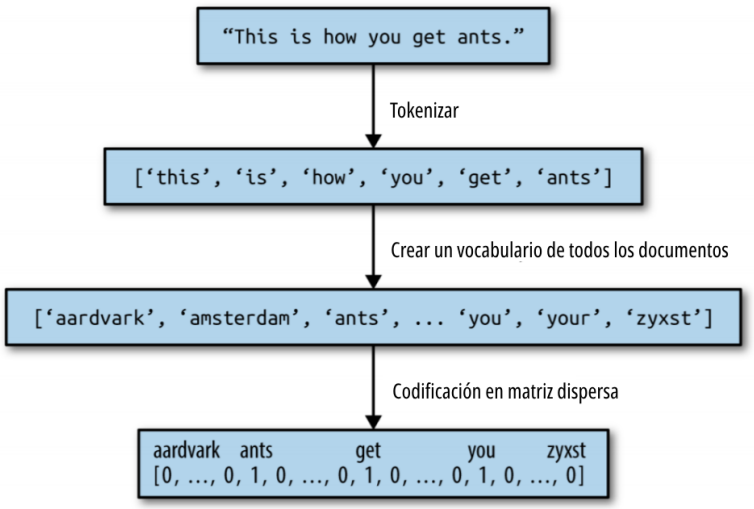
$$D = \{W_{D1}, W_{D2}, \dots, W_{Di}, \dots, W_{Dn}\} \tag{2.1}$$

Aquí, W_{Di} representa el peso del término W_i, siendo esta la i-ésima palabra en el documento D. Los pesos para cada W_{Di} pueden ser calculados mediante diferentes técnicas de ponderación, como lo son: Bolsa de Palabras, TF-IDF y TF-IGM. Estas técnicas se describen a continuación.

Bolsa de Palabras

La bolsa de palabras (BoW, por sus siglas en inglés de *Bag of Words*) es una de las formas más simples para transformar texto en representaciones numéricas, la cual sirve como punto de partida para técnicas más complejas como TF-IDF y TF-IGM. BoW consiste en recuperar las palabras únicas que aparecen en todo el corpus de documentos, formando un vocabulario, descartando cualquier estructura gramatical, para, posteriormente, por cada documento contar cuántas veces aparece cada palabra presente en el vocabulario [Müller y Guido 2017]. En la Figura **2.3**, se muestra un ejemplo de la representación vectorial de documentos utilizando BoW.

Figura 2.3
Representación del proceso para construir una bolsa de palabras



Nota. Fuente [Kharwal 2020].

El origen del nombre de BoW proviene del hecho de que al descartar la estructura y contar únicamente las ocurrencias de palabras, nos lleva a imaginar mentalmente el texto como una "bolsa". De este modo, como se observa en la Figura **2.3** para un documento D vectorizado mediante BoW, el peso de cada palabra corresponde a la frecuencia con la que aparece en ese documento [Sarkar 2016].

TF-IDF

El modelo TF-IDF (*Term Frequency–Inverse Document Frequency*) es una técnica no supervisada que pondera el conteo de palabras o términos para darle mayor peso a aquellas con una mayor frecuencia en documentos específicos, pero que no aparecen en la gran mayoría de documentos del corpus [Sarkar 2016]. Si una palabra aparece a menudo en un documento específico, pero no en muchos otros, es probable que sea muy descriptiva del contenido de ese documento.

Matemáticamente, TF-IDF es el producto de dos factores, y se representa como:

$$tfidf = tf \times idf \tag{2.2}$$

Donde, para un término t, se tiene que calcular:

$$tf(w, D) = f_{w_D} \tag{2.3}$$

$$idf(t) = 1 + \log(\frac{C}{1 + df(t)}) \tag{2.4}$$

Aquí, f_{w_D} denota la frecuencia para la palabra w en el documento D, que se convierte en la frecuencia de término, tf. Por otro lado, $idf(t)$ representa el valor idf para el término t, C representa el conteo del número total de documentos en el corpus, y $df(t)$ es la frecuencia del número de documentos en los que el término t está presente.

TF-IGM

TF-IGM (*Term Frequency–Inverse Gravity Moment*) es una técnica de ponderación de términos similar a TF-IDF, pero con la diferencia de que está diseñada para mejorar sus limitaciones en tareas supervisadas como la clasificación, ya que TF-IDF no toma en cuenta la distribución de clases en el corpus [Chen et al. 2016].

Esta técnica introduce el factor IGM, que mide la capacidad de un término de distinguir clases en un corpus, utilizando como inspiración el concepto de momento gravitacional proveniente de la física, para evaluar la no uniformidad o concentración de la distribución inter-clase de los términos [Polpinij y Luaphol 2021].

El factor IGM puede expresarse como:

$$igm(t) = \frac{df_{t_1}}{\sum_{r=1}^{m} df_{t_r} \times r} \tag{2.5}$$

Aquí, $igm(t)$ denota el momento gravitacional inverso de la distribución inter-clase del

término t, en el cual, m es el número total de clases y r representa un rango; por lo que, df_{t_r} se refiere al número de documentos que contienen el término t en la r-ésima clase, ordenados de forma descendente. Un término que aparece mayormente en una clase específica tendrá un valor IGM más alto, lo que indica una mayor capacidad de diferenciación.

Finalmente, TF-IGM se calcula como:

$$tfigm = tf \times (1 + \lambda \times igm) \tag{2.6}$$

Aquí, λ es un coeficiente ajustable que controla la contribución relativa del factor IGM en el peso final. Este coeficiente tiene un valor por defecto de 7.0, pero puede ajustarse a un valor en el rango de $[5.0, 7.0]$.

Selección de términos relevantes en el Modelo de Espacio Vectorial

Uno de los principales problemas en la clasificación de textos es la alta dimensionalidad generada por las técnicas de vectorización basadas en el VSM. Este fenómeno, conocido comúnmente como la *maldición de la alta dimensionalidad*, se refiere a la tendencia de los conjuntos de datos con un gran número de atributos o características a volverse dispersos. Esta dispersión dificulta el proceso de aprendizaje, incrementa el riesgo de sobreajuste y reduce la fiabilidad de las predicciones [Géron 2019].

Como se ha mencionado, los VSMs están compuestos por términos o palabras únicas que aparecen en todos los documentos que conforman el corpus; sin embargo, algunos de estos términos podrían no tener un impacto significativo en el proceso de clasificación. Por lo tanto, es necesario priorizar solo aquellas características que sean verdaderamente relevantes. La selección de características es un método que permite identificar y elegir un subconjunto relevante de características de un conjunto de datos. El objetivo de la selección de características es mejorar el rendimiento general del modelo y reducir su complejidad computacional, así como facilitar su interpretación [Meesad et al. 2011].

Uno de los métodos usados para realizar una selección de características en los VSMs, es la prueba estadística de chi-cuadrado (χ^2). Esta prueba se utiliza para medir la falta de

independencia entre un término t y una clase c, que se compara con la distribución del mismo nombre con cierto grado de libertad [Wibowo Haryanto et al. 2018]. La fórmula de χ^2 para realizar una selección de características es la siguiente:

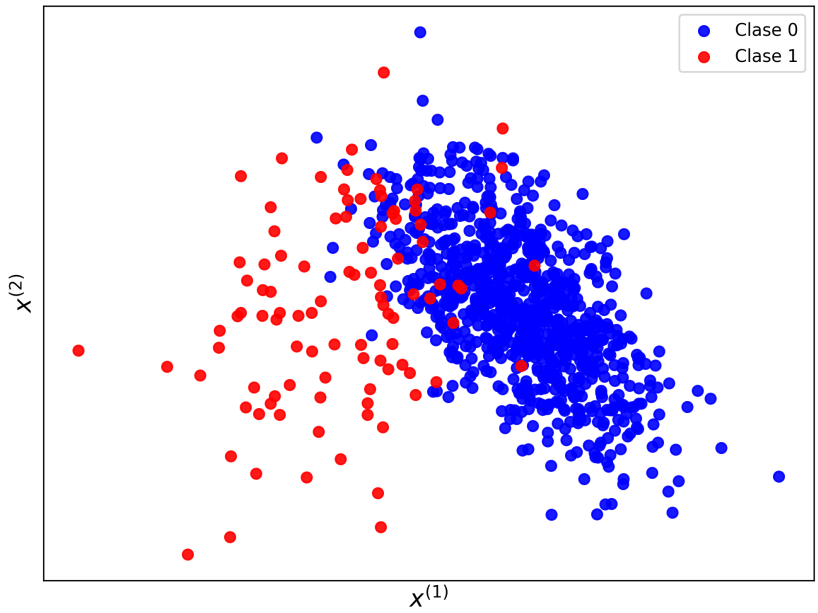
$$X_{(c,t)} = \frac{D(AJ - BI)^2}{(A + I)(B + J)(A + B)(I + J)} \tag{2.7}$$

Aquí, D es el número total de documentos; A, el número de documentos de la clase c que contienen t; B el número de documentos que contienen t, pero no pertenecen a la clase c; I, el número de documentos de la clase c que no contienen t; y, por último, J, el número de documentos de otras clases que no contienen t. El resultado de aplicar χ^2 es un conjunto reducido de características compuesto por los términos que tienen una mayor dependencia estadística con las clases presentes en el conjunto de datos.

2.2 Balanceo de clases

Además de considerar el preprocesamiento y la extracción de características para el análisis de textos, es necesario considerar el balanceo de clases. En muchos problemas del mundo real, es común encontrar la situación donde las instancias de una clase están poco representadas en el conjunto de datos, denominándose como la minoritaria, en comparación con aquella clase que concentra la mayoría de las instancias, la mayoritaria. Esto se puede ver representado en la Figura **2.4**, que se muestra un conjunto de datos sintético con un total de 1000 instancias que presenta una relación de desbalance de 1:9. Es decir, por cada instancia de la clase minoritaria (clase 1) existen 9 instancias de la mayoritaria (clase 0), siendo un total de 900 instancias en la clase 0 y 100 instancias en la clase 1.

Figura 2.4
Visualización de un conjunto de datos desbalanceado



Nota. Conjunto de datos desbalanceado, con dos clases y dos características, generado sintéticamente. Fuente propia.

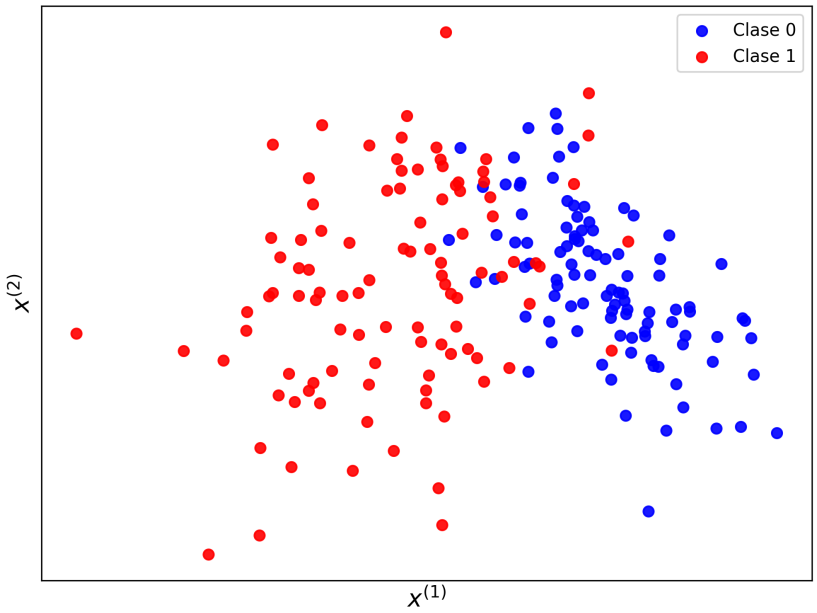
Estos conjuntos de datos desbalanceados impactan de manera negativa el desempeño de los algoritmos de aprendizaje automático, ya que tenderán a favorecer a la clase mayoritaria, resultando en modelos de clasificación con poca sensibilidad a la clase minoritaria [Burkov 2023].

No obstante, obtener más datos para resolver este problema no siempre es posible, esto debido a diversas situaciones y limitaciones. Por lo mismo, se recurre a utilizar técnicas basadas en remuestreo o *resampling* [Dharma y Saragih 2022; Gazali Mahmud et al. 2023; Satriaji y Kusumaningrum 2018], para balancear la distribución de ejemplos en las clases para crear modelos de clasificación que no favorezcan únicamente a la clase mayoritaria, sino que también sean capaces de identificar eficazmente las instancias pertenecientes a la clase minoritaria.

2.2.1 Submuestreo

El submuestreo (*undersampling* en inglés) es el enfoque que balancea el conjunto de datos seleccionando un subconjunto de instancias en la clase mayoritaria que iguale el número de instancias en la minoritaria. Sus principales ventajas son la rapidez para su implementación y el bajo coste computacional, ya que se trabaja con un conjunto de datos más pequeño. Sin embargo, al descartar instancias es posible que se produzca la pérdida de información que podría resultar valiosa para el entrenamiento de los modelos [Fernández et al. 2018].

Figura 2.5
Visualización del resultado de aplicar RUS a un conjunto de datos desbalanceado



Nota. Fuente propia.

Entre las técnicas más simples, pero ampliamente utilizadas de tipo *undersampling*, se encuentra el Random Undersampling (RUS). Esta técnica no heurística busca balancear la distribución de clases mediante la eliminación de manera aleatoria de instancias de la clase mayoritaria hasta que la relación de desbalance, o proporción de instancias entre clases, esté equilibrada. A manera de ejemplo, aplicando RUS al conjunto de datos sintéticos presentado

en la Figura **2.4**, obtenemos como resultado el conjunto de datos visualizado en la Figura **2.5**, el cual redujo la cantidad de instancias en la clase mayoritaria a 100 para igualar a la minoritaria, presentando entonces una relación de desbalance de 1:1.

2.2.2 Sobremuestreo

El sobremuestreo (*oversampling* en inglés) es el enfoque que busca replicar o generar nuevas instancias sintéticas en la clase minoritaria con el fin de aumentar su importancia, así como igualar la cantidad de instancias con las presentes en la clase mayoritaria [Burkov 2023; Fernández et al. 2018].

La técnica más simple y que sirve de punto de partida a técnicas más complejas, es Random Oversampling (ROS). Esta técnica no heurística, contraria a RUS, replica de manera aleatoria instancias de la clase minoritaria con el fin de balancear la distribución de clases. No obstante, el uso de ROS puede aumentar la probabilidad de sobreajuste, dado que se realiza una replicación de copias exactas y puede llevar a que un clasificador aparente tener una buena generalización, pero que en realidad solo cubre ciertos casos específicos [Fernández et al. 2018]. A continuación se describen técnicas más avanzadas que buscan mitigar este riesgo.

SMOTE

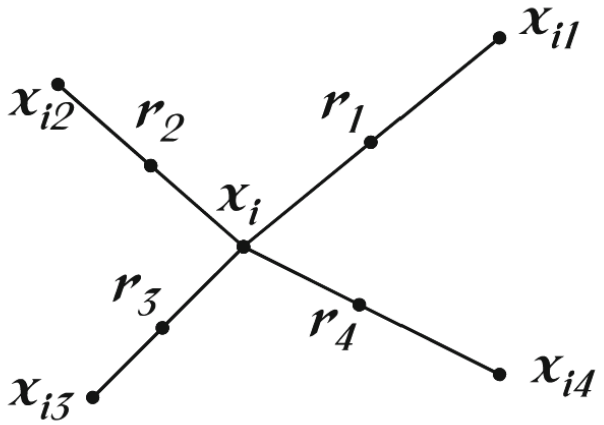
SMOTE (*Synthetic Minority Oversampling TEchnique*) es una técnica de *oversampling* que busca balancear un conjunto de datos mediante la generación de instancias sintéticas con base en la información presente de las instancias pertenecientes a la clase minoritaria. Estas nuevas instancias se crean mediante la interpolación entre varias instancias que están cerca unas de otras [Fernández et al. 2018].

El procedimiento consiste en definir el número N de instancias sintéticas a generar por instancia presente en la clase minoritaria. Este número, por defecto, se define como el número que permite alcanzar una relación de 1:1. SMOTE utiliza kNN para la generación, en la que se selecciona aleatoriamente una instancia minoritaria y se identifican sus k vecinos

más cercanos, donde por defecto: $k = 5$. De estos, se eligen al menos n vecinos (que pueden repetirse) para obtener su distancia de diferencia respecto al vector de características de la instancia. Posteriormente, se multiplica esta distancia por un valor aleatorio en el rango de $[0, 1]$ y se suma al vector de características de la instancia, lo que finalmente produce la nueva instancia sintética. Este procedimiento se repite hasta que el conjunto de datos esté balanceado [Fernández et al. 2018; Mujahid et al. 2024].

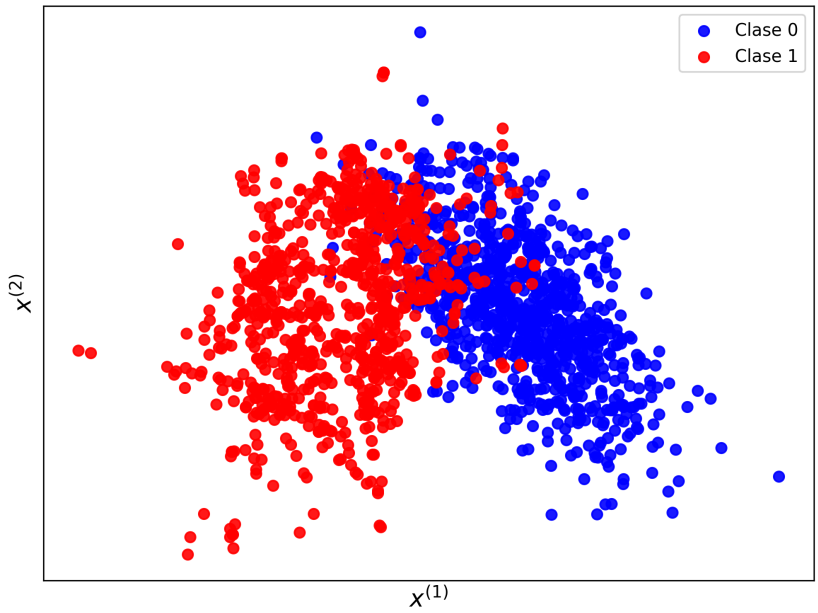
A manera de ejemplo, en la Figura **2.6** se ilustra el proceso para generar cuatro instancias sintéticas r_1, r_2, r_3, r_4 a partir de una instancia x_i y sus cuatro vecinos $x_{i1}, x_{i2}, x_{i3}, x_{i4}$ obtenidos mediante kNN. De manera similar, en la Figura **2.7** observamos el resultado de aplicar SMOTE al conjunto de datos presentado en la Figura **2.4**.

Figura 2.6
Creación de una instancia sintética con SMOTE



Nota. Fuente [Fernández et al. 2018].

Figura 2.7
Visualización del resultado de aplicar SMOTE a un conjunto de datos desbalanceado



Nota. Fuente propia.

DEBOHID

DEBOHID, un enfoque de *oversampling* basado en el evolutivo diferencial para conjuntos de datos altamente desbalanceados (del inglés: *a Differential Evolution Based Oversampling approach for Highly Imbalanced Datasets*), es una técnica que genera nuevas instancias sintéticas para la clase minoritaria usando como base la estrategia *DE/rand/1/bin* procedente de la metaheurística de Evolutivo Diferencial [Kaya et al. 2021].

El Evolutivo Diferencial (*DE*, por sus siglas en inglés de *Differential Evolution*) es un algoritmo evolutivo basado en poblaciones propuesto para resolver problemas de optimización sobre dominios continuos, el cual se inspira en la evolución de las especies y la selección natural. Sus tres operadores principales son: mutación, cruza y selección, mediante los cuales las poblaciones van *mutando* para que sus individuos (matemáticamente representados como vectores) se vuelvan soluciones óptimas para resolver un problema [Coello et al. 2007].

DE presenta variantes que corresponden a la forma en que se introduce la mutación. Dichas variantes se conocen como *estrategias*, siendo la estándar *DE/rand/1/bin* donde: *rand* indica que los individuos con los que se calculará el vector de mutación para cada individuo de la población se seleccionarán de manera aleatoria; 1 es la cantidad de diferencias de vectores a utilizar para calcular el vector de mutación; y, finalmente, *bin* significa que la cruza entre el individuo y su correspondiente vector de mutación se realizará usando cruza binomial.

La creación del vector de mutación v_i correspondiente para el individuo x_i utilizando la estrategia descrita se realiza mediante la siguiente ecuación:

$$v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \tag{2.8}$$

Aquí, $x_{(r_1,r_2,r_3)}$ son los individuos seleccionados aleatoriamente diferentes de x_i, $(x_{r_2} - x_{r_3})$ es la diferencia de vectores y F es un factor que escala la influencia de la diferencia, que puede tomar cualquier valor en el rango de $(0, 2)$. Por otro lado, el proceso de cruza que construye el nuevo individuo u_i se realiza utilizando:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{si } rand(0,1) < CR \\ x_{i,j} & \text{si } rand(0,1) \geq CR \end{cases} \tag{2.9}$$

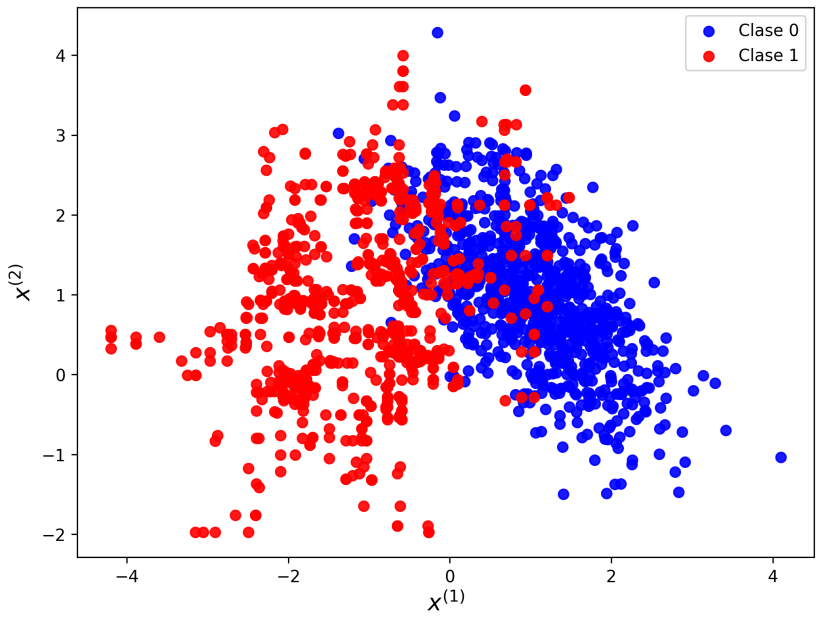
Aquí, CR es la probabilidad de cruza que controla la selección del vector de mutación o la prevalencia del individuo, tomando valores en el rango $(0, 1)$. Esta operación se aplica para cada dimensión j en ambos vectores.

Por su parte, el procedimiento que DEBOHID utiliza empieza determinando las constantes F, CR y NOS, siendo este último el número de instancias sintéticas a generar. Así también, se establecen los límites superior e inferior de las características presentes en el conjunto de datos. Entonces, por cada ejemplo a generar se selecciona una instancia de la clase minoritaria, de la cual, mediante kNN se obtienen sus 3 vecinos más cercanos. Teniendo estos datos, se crea su correspondiente vector de mutación utilizando la ecuación **2.8** y se combinan conforme lo especificado en la ecuación **2.9**. Finalmente, se verifica que

los valores generados para las características estén dentro de los límites; en caso de excederlos, se ajustan al límite correspondiente. Así, el vector resultante es la nueva instancia sintética. El procedimiento se repite hasta generar la cantidad especificada en NOS [Kaya et al. 2021].

A manera de ejemplo, en la Figura **2.8** se observa el resultado de aplicar DEBOHID al conjunto de datos presentado en la Figura **2.4**.

Figura 2.8
Visualización del resultado de aplicar DEBOHID a un conjunto de datos desbalanceado



Nota. Fuente propia.

2.2.3 Aumento de datos basado en manipulación textual

El uso de técnicas como SMOTE o DEBOHID que trabajan sobre características presentes en VSMs genera nuevas instancias sintéticas que no tienen un significado lógico en lenguaje natural, por lo que se vuelve información abstracta y difícil de interpretar por un humano, contrario a tener los datos generados en texto comprensible. [Wei y Zou 2019] propusieron un método de aumento de datos conocido como EDA (*Easy Data Augmenta-*

tion Techniques for Boosting Performance on Text Classification Tasks) que busca mejorar el rendimiento en tareas de clasificación de textos usando operaciones que modifican directamente los datos textuales y no sobre vectores de características para generar las nuevas instancias sintéticas.

El proceso que EDA sigue para realizar la generación comienza definiendo n_{aug} que es la cantidad de instancias sintéticas generadas por cada ejemplo original en el conjunto de datos. Entonces, en cada generación, se selecciona de manera aleatoria y se aplica una de las siguientes operaciones:

1. **Reemplazo por Sinónimo.** Se seleccionan aleatoriamente n palabras que no sean palabras vacías y se reemplazan con uno de sus sinónimos que, de igual forma, se eligen aleatoriamente.
2. **Inserción Aleatoria.** De una palabra elegida aleatoriamente se selecciona uno de sus sinónimos de la misma forma y se inserta en una posición aleatoria en el texto. Esta operación se repite n veces.
3. **Intercambio Aleatorio.** Se seleccionan dos palabras de forma aleatoria y se intercambian sus posiciones. Esta operación se repite n veces.
4. **Eliminación Aleatoria.** Aleatoriamente, se elimina cada palabra del texto dada una probabilidad p.

Para compensar el hecho de que los textos con más palabras son propensos a absorber más ruido que aquellos con menos palabras, se ajusta el número n de palabras modificadas en las tres primeras operaciones en función de su longitud l, utilizando la fórmula $n = \alpha l$, donde α es un parámetro que indica el porcentaje de palabras que se modificarán. En el caso de la última operación, se usa $p = \alpha$ como la probabilidad de eliminar una palabra. En la Tabla **2.1** se muestran ejemplos de la aplicación de las operaciones.

Tabla 2.1
Ejemplos de textos generados aplicando EDA

Operación	Texto
Ninguna	La vida es un hermoso viaje lleno de aprendizajes y momentos inolvidables.
1	La vida es un **precioso** viaje lleno de **enseñanzas** y momentos inolvidables.
2	La vida es un hermoso **inspirador** viaje lleno de aprendizajes y momentos inolvidables.
3	La vida es un hermoso **inolvidables** lleno de aprendizajes y momentos **viaje**.
4	La vida es un hermoso viaje lleno de momentos inolvidables.

Nota. Fuente [Wei y Zou 2019].

2.3 Aprendizaje Automático

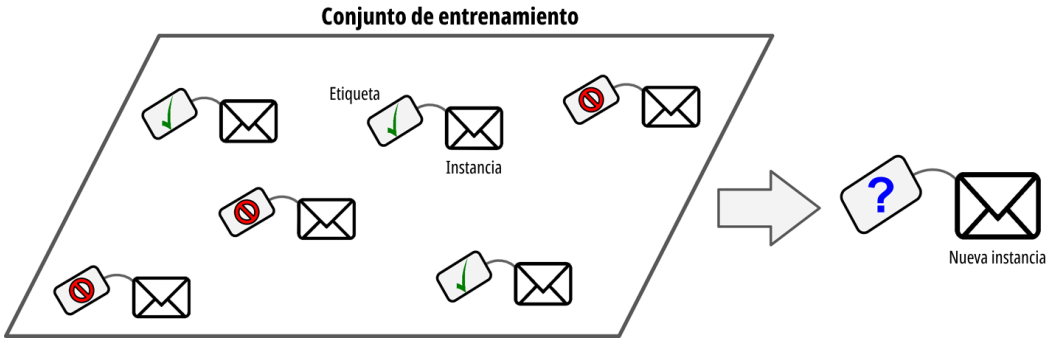
El Aprendizaje Automático (ML, por sus siglas en inglés de *Machine Learning*) es un subcampo de las ciencias de la computación que se encarga de crear algoritmos que, mediante una colección de instancias o datos de algún fenómeno, dan a las computadoras la habilidad de aprender a resolver problemas sin la necesidad de ser explícitamente programadas [Burkov 2023; Géron 2019]. Entre todos los algoritmos disponibles, aquellos que aprenden a partir de datos etiquetados, es decir, datos que incluyen un resultado esperado asociado a cada instancia, pertenecen a la categoría de aprendizaje supervisado [Müller y Guido 2017].

El aprendizaje supervisado, uno de los tipos más usados de ML, es utilizado en situaciones donde queremos predecir cierto resultado a partir de una entrada dada. Este tipo de algoritmos requiere para su entrenamiento conjuntos de datos con características numéricas o categóricas que describen un fenómeno y que incluyan también la solución o resultado esperado. Dependiendo del tipo de variable que se desea predecir, estas tareas pueden dividirse en problemas de regresión, donde se estima un valor continuo, o de clasificación, donde se asigna una categoría discreta. El objetivo de estos algoritmos es hacer prediccio-

nes precisas para datos nuevos, nunca antes vistos en su conjunto de entrenamiento [Müller y Guido 2017].

Entre las tareas de clasificación, una de las más representativas es la detección de correos electrónicos o emails no deseados. Este problema consiste en detectar si un email es spam o, por el contrario, no lo es, representando un problema de clasificación binaria. Así, el algoritmo de ML utiliza varias instancias de emails etiquetados como *spam* o *no spam*, como se observa en la Figura **2.9**, con la finalidad de generar un modelo capaz de clasificar nuevos emails de manera eficaz [Géron 2019].

Figura 2.9
Ejemplo de etiquetado de instancias para detección de spam



Nota. Fuente [Géron 2019].

Para optimizar el desempeño de los modelos generados, es necesario ajustar los hiperparámetros correspondientes de los algoritmos.

2.3.1 Regresión Logística

La Regresión Logística (LR, por sus siglas en inglés de *Logistic Regression*) es un modelo estadístico de tipo paramétrico usado para estimar la probabilidad de que una instancia pertenezca a una clase en particular, en función de las características de entrada [Géron 2019].

La LR se basa en una combinación lineal de las características de entrada, transforma-

da mediante la función logística (o sigmoide), que asegura que la salida sea un valor en el intervalo $[0, 1]$. Matemáticamente, la probabilidad de que una instancia x pertenezca a la clase denominada como 1, se expresa como:

$$f(x) = \frac{1}{1 + e^{-(w \cdot x + b)}} \tag{2.10}$$

Aquí, w es un vector de pesos, x es el vector de características de entrada, b es el sesgo y e es el número de Euler.

La salida del modelo puede interpretarse como que, si la probabilidad estimada es mayor a cierto umbral definido (generalmente 50 %) entonces la predicción es que pertenece a la clase 1; por el contrario, si es menor, la predicción es que pertenece a la clase 0. Esta particularidad hace de este modelo un clasificador binario.

Para entrenar el modelo, en lugar de minimizar una función de pérdida como en la regresión lineal, se maximiza la verosimilitud del conjunto de entrenamiento. Esto implica ajustar w y b para que las predicciones del modelo sean lo más cercanas posible a las etiquetas reales [Burkov 2023].

2.3.2 Clasificador Bayesiano Ingenuo

El Clasificador Bayesiano Ingenuo (NB, por sus siglas en inglés de *Naïve Bayes*) es un algoritmo probabilístico basado en el Teorema o Regla de Bayes, que hace suposiciones ingenuas sobre que las características presentes en el conjunto de datos son estadísticamente independientes dado el valor de clase. Este supuesto se considera con el fin de simplificar el cálculo de las probabilidades para predecir la clase de una instancia dada según su conjunto de datos de entrenamiento [Chowdhary 2020].

La razón de esto es que, la aplicación directa del Teorema de Bayes no es práctica, dado que, la existencia de dependencia entre características obliga a calcular probabilidades conjuntas entre ellas que aumentan significativamente la complejidad y necesidad de una gran cantidad de datos para que el clasificador pueda aprender estas dependencias con

exactitud. Este supuesto es el motivo de la procedencia de su nombre.

El algoritmo de NB se entrena estimando $P(x_j \mid y_k)$ y $P(y_k)$, que son la probabilidad condicional de que una característica x_j tome un valor específico dado un valor de clase y_k, y la probabilidad a priori de cada clase, respectivamente. Estos valores se calculan a partir de las frecuencias relativas en el conjunto de entrenamiento. Finalmente, la probabilidad conjunta de todas las características $(x_1, \ldots, x_j, \ldots, x_n)$, siendo esta el producto de las probabilidades individuales, se obtiene mediante la siguiente fórmula:

$$\hat{y} = \arg\max_{y_i} \left[P(y_i) \prod_{j=1}^{n} P(x_j \mid y_i) \right] \tag{2.11}$$

Donde $\hat{y}$ representa la clase predicha para una instancia dada. De esta manera, el algoritmo asigna a la instancia la clase y_i cuya combinación de probabilidad a priori y probabilidad condicional de las características observadas resulta en el valor más alto.

2.3.3 Máquina de Soporte Vectorial

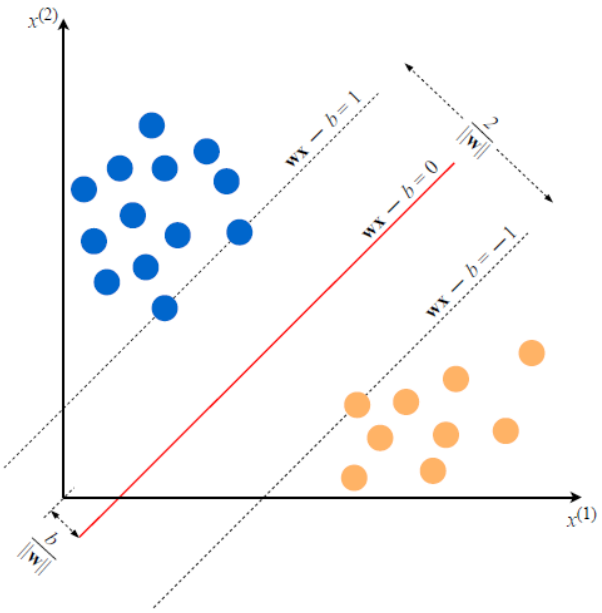
La Máquina de Soporte Vectorial (SVM, por sus siglas en inglés de *Support Vector Machine*) es un clasificador lineal no probabilístico en el cual los datos se representan mediante puntos en un espacio que puede ser multidimensional; dependiendo de la cantidad de características presentes. Estos datos serán separados en dos clases o categorías durante el entrenamiento mediante un hiperplano, de tal forma que los datos de cada categoría están divididos por un margen claro y lo más amplio posible [Chowdhary 2020].

El margen, ilustrado en la Figura **2.10**, corresponde a la distancia entre los puntos más próximos de cada clase y el hiperplano. Su maximización contribuye a mejorar la capacidad del modelo para generalizar y clasificar correctamente nuevas instancias [Burkov 2023]. La ecuación del hiperplano se define como:

$$wx - b = 0 \tag{2.12}$$

Aquí, w es un vector de pesos, x es el vector de características del ejemplo y b es el sesgo.

Figura 2.10
Representación de un modelo de SVM en un espacio de dos dimensiones



Nota. Fuente [Burkov 2023].

El entrenamiento del modelo tiene como objetivo determinar los valores óptimos de w y b que definan el hiperplano. SVM resuelve esto como un problema de optimización, buscando minimizar los errores de clasificación mientras maximiza el margen. Para ello, identifica el hiperplano que se encuentra a la mayor distancia posible de los puntos más cercanos de cada clase, denominados vectores de soporte, lo que da origen a su nombre.

Para realizar la clasificación de una instancia nueva, el algoritmo calcula la función:

$$y = sign(wx - b) \tag{2.13}$$

Aquí, $sign$ es un operador o función que toma cualquier valor numérico como entrada y

regresa $+1$ si es positivo y -1 si es negativo, indicando la clase asignada según la predicción del clasificador.

2.4 Evaluación de rendimiento

Una vez construido un modelo mediante los datos en el conjunto de entrenamiento, debemos comprobar su desempeño al realizar la tarea en cuestión. Para esto, se hace uso del conjunto de prueba, el cual posee instancias que el modelo nunca vio durante su entrenamiento, por lo que, si realiza un buen trabajo prediciendo las etiquetas de las instancias en este conjunto, se dice que el modelo tiene una buena *generalización* [Müller y Guido 2017]. No obstante, alcanzar una buena *generalización* no siempre es posible, y las razones de esto van desde la calidad de los datos hasta las capacidades del modelo utilizado.

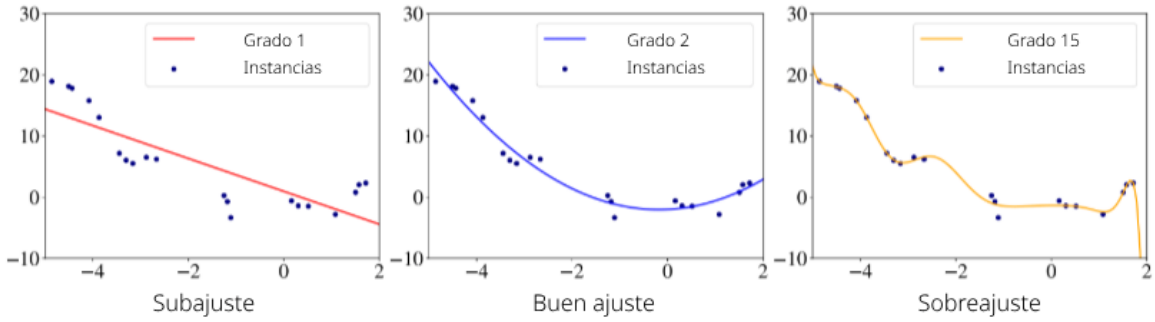
Se dice que un modelo alcanzó un sobreajuste (*overfitting*, en inglés) cuando es capaz de predecir muy bien los datos de entrenamiento, pero es deficiente prediciendo los datos de prueba. Esto indica una alta varianza, lo que significa que el modelo es sensible a pequeñas variaciones en los datos de entrenamiento. En este caso, es probable que, al tratar de predecir correctamente todas las instancias en el conjunto de entrenamiento, el modelo haya aprendido no solo los patrones relevantes en los datos, sino también el ruido presente.

Por otro lado, encontramos el subajuste (*underfitting*, en inglés), el cual ocurre cuando un modelo se equivoca la mayoría de las veces aun prediciendo las etiquetas de los datos en el conjunto de entrenamiento, por lo que se dice que el modelo tiene un alto sesgo. Las principales causas para que se dé un *underfitting* son que el modelo sea demasiado simple para los datos o que las características utilizadas no son lo suficientemente informativas sobre el fenómeno a predecir [Burkov 2023].

Podemos ver un ejemplo de cada caso en la Figura **2.11**. El subajuste se representa con un modelo lineal que no ajusta bien los datos; un buen ajuste se muestra con un modelo cuadrático que captura bien los patrones de los datos, que resulta en una buena generalización; y el sobreajuste se ilustra con un modelo polinómico de grado elevado que se ajusta demasiado bien a los datos de entrenamiento. Para medir esto, en el área de ML se hace uso

de distintas técnicas de evaluación, con el fin de determinar su capacidad de generalización y su eficacia en resolver problemas del mundo real.

Figura 2.11
Ejemplos de subajuste, buen ajuste y sobreajuste



Nota. Fuente [Burkov 2023].

2.4.1 Métricas

Las métricas de evaluación son aquellas que permiten cuantificar el desempeño de los modelos en distintos aspectos, lo que facilita la comparación entre diferentes técnicas de preprocesamiento y algoritmos de aprendizaje utilizados para construirlos. En un problema de clasificación binario, las instancias de los datos son comúnmente clasificadas como positivas o negativas, donde una etiqueta positiva indica la presencia de un fenómeno o anomalía, mientras que una negativa se considera como algo que no está fuera de lo normal [Rainio et al. 2024].

Una predicción utilizando un etiquetado binario tiene cuatro posibles designaciones:

- **True Positive (TP)**. Predicción correcta de un resultado positivo.
- **True Negative (TN)**. Predicción correcta de un resultado negativo.
- **False Positive (FP)**. Instancia negativa clasificada incorrectamente como positiva.
- **False Negative (FN)**. Instancia positiva clasificada incorrectamente como negativa.

La matriz de confusión se presenta como una herramienta para registrar estos valores. En el caso binario, tenemos una matriz de 2×2 donde las filas representan las etiquetas reales y las columnas, las predicciones del modelo. Cada celda muestra la cantidad de instancias que pertenecen a la clase indicada en la fila que fueron clasificadas como la clase indicada en la columna [Müller y Guido 2017]. En la Figura **2.12** se muestra un ejemplo de matriz de confusión.

Figura 2.12
Representación de una matriz de confusión para clasificación binaria

		Verdadero	
		Positivos	**Negativos**
Predicción	**Positivos**	TP	FP
	Negativos	FN	TN

Nota. Fuente propia.

Estas observaciones pueden ser usadas para calcular diversas métricas para la evaluación de un modelo, que se describen a continuación.

Precisión

La precisión, o *Precision* en inglés, indica qué proporción de las instancias clasificadas como positivas (TP y FP) son verdaderamente positivas. Dicho de otra forma, la precisión evalúa que tan confiable es un modelo al clasificar instancias como positivas. Su fórmula es:

$$Prec. = \frac{TP}{TP + FP} \tag{2.14}$$

Sensibilidad

La sensibilidad, conocida como *Sensitivity*, *Recall* o *True Positive Rate* en inglés, evalúa la proporción de instancias realmente positivas (TP y FN) que han sido correctamente identificadas como positivas (TP). Es útil cuando se busca maximizar la detección de verdaderos positivos y minimizar los falsos negativos. Su fórmula es:

$$Sens. = \frac{TP}{TP + FN} \tag{2.15}$$

Especificidad

La especificidad, o *Specificity* en inglés, evalúa la proporción de instancias realmente negativas (TN y FP) que han sido correctamente identificadas como negativas (TN). Sirve como complemento de la sensibilidad, destacando la correcta clasificación de los negativos y la disminución de falsos positivos. Su fórmula es:

$$Esp. = \frac{TN}{TN + FP} \tag{2.16}$$

F-Score

El F-Score, también conocido como F1, es una métrica combinada que se define como la media armónica entre la precisión y la sensibilidad. Al considerar ambas métricas, resulta más adecuada que la exactitud en escenarios con conjuntos de datos desbalanceados. Su fórmula es:

$$F1 = \frac{2 \times Prec. \times Sens.}{Prec. + Sens.} \tag{2.17}$$

G-Mean

La media geométrica, conocida como *Geometric Mean*, *G-Mean*, en inglés, se define como la raíz enésima del producto de la sensibilidad de cada clase. Esta métrica indica que, aunque un modelo pueda clasificar correctamente la mayoría de las instancias, un bajo rendimiento en la predicción de instancias de la clase minoritaria se reflejará en un valor reducido de *G-Mean*. Por ello, resulta adecuada en escenarios con conjuntos de datos desbalanceados.

Dada la sensibilidad $Sens._i$ obtenida para cada clase i en un conjunto con n clases, G-Mean se calcula como:

$$GMean = \sqrt[n]{\prod_{i=1}^{n} Sens._i} \tag{2.18}$$

En el caso binario, como la sensibilidad de la clase negativa es la especificidad, su fórmula es:

$$GMean = \sqrt{Sens. \times Esp.} \tag{2.19}$$

Exactitud Balanceada

La exactitud tradicional puede no ser una métrica adecuada cuando las clases están desbalanceadas. En estos casos, el modelo puede obtener una alta exactitud simplemente prediciendo la clase mayoritaria con mayor frecuencia, sin necesariamente recuperar correctamente las instancias de la clase minoritaria. En su lugar, se utiliza la Exactitud Balanceada, o *Balanced Accuracy* (BA) como se conoce en inglés. La BA es la media aritmética de las sensibilidades por clase.

Dada la sensibilidad $Sens._i$ obtenida para cada clase i en un conjunto con n clases, la BA se calcula como:

$$BA = \frac{1}{n} \sum_{i=1}^{n} Sens._i \tag{2.20}$$

En el caso binario, como la sensibilidad de la clase negativa es la especificidad, su fórmula es:

$$BA = \frac{1}{2}(Sens. + Esp.) \tag{2.21}$$

Promedios de Métricas

Una manera comúnmente utilizada para evaluar modelos entrenados con conjuntos de datos desbalanceados, especialmente cuando se emplean métricas como precisión, sensibilidad y F1, es utilizar sus promedios. En particular, el promedio Macro (PM) calcula la media aritmética de la métrica utilizada por clase, sin ponderarla. Esto asigna el mismo peso a todas las clases, independientemente de su tamaño. Dada una métrica M y sus correspondientes i resultados para las n clases, el PM se define como:

$$PM = \frac{1}{n} \sum_{i=1}^{n} M_i \tag{2.22}$$

2.4.2 Validación cruzada

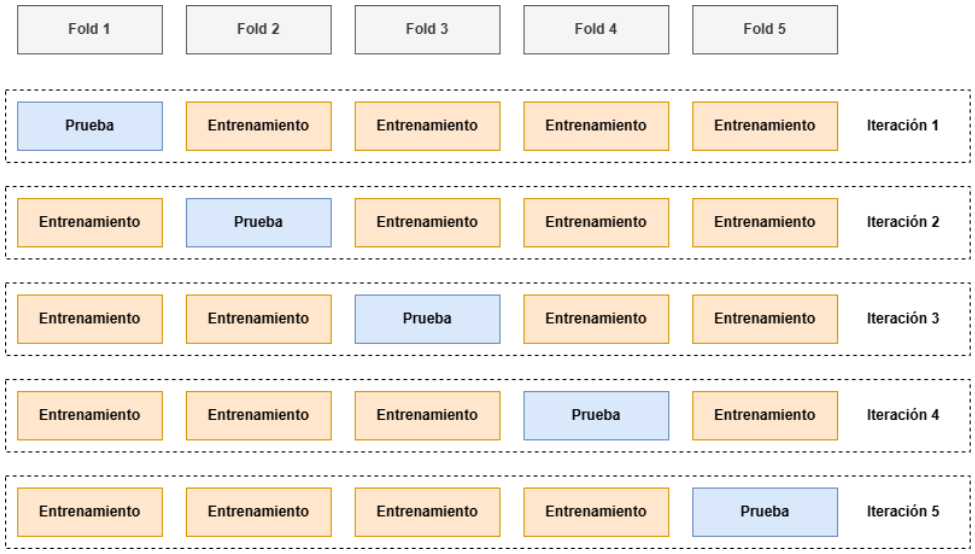
La validación cruzada es un método estadístico para evaluar la capacidad de generalización de un modelo de aprendizaje mediante la utilización de diferentes segmentos en los datos como conjuntos de entrenamiento y prueba que es más estable y exhaustivo que el uso de una división simple de los datos en los conjuntos mencionados [Müller y Guido 2017]. Este método permite realizar simulaciones para observar cómo podría comportarse

un modelo al realizar predicciones sobre datos nunca vistos durante su entrenamiento.

De las técnicas existentes destaca *K-Fold Cross-Validation*, para la cual su procedimiento consiste en, primero, definir el número de K cortes para dividir los datos en varios subconjuntos del mismo tamaño, llamados *folds*. Este valor de K también denota la cantidad de modelos a entrenar, por lo que, en el caso de $K = 5$, los datos se dividen aleatoriamente en cinco partes iguales: $\{F_1, F_2, F_3, F_4, F_5\}$, cada una con el 20 % del total de los datos.

Luego, de manera iterativa, se entrenan los K modelos asignando en cada iteración uno de los *folds* como conjunto de prueba, mientras que los restantes se utilizan para el entrenamiento. Este procedimiento se repite hasta que cada *fold* ha sido usado como conjunto de prueba una vez. Un ejemplo de esto se puede ver en la Figura **2.13**.

Figura 2.13
División de datos usando K-Fold con $K = 5$



Nota. Fuente propia.

Finalmente, se calcula el valor de la métrica de evaluación de rendimiento de interés en cada conjunto de prueba y se calcula el promedio de los resultados obtenidos, que representa el desempeño del modelo [Burkov 2023]. Adicionalmente, en lugar de considerar únicamente el promedio, es posible utilizar intervalos de confianza para estimar la variabilidad de los resultados y proporcionar una evaluación más robusta del desempeño.

Por otro lado, cuando se trabaja con conjuntos de datos que presenten desbalance de clases, se utiliza una variación llamada *Stratified K-Fold Cross-Validation* (SKCV), en la cual, su principal diferencia respecto a *K-Fold* es que SKCV realiza un muestreo estratificado para generar *folds* que contengan una proporción representativa de cada clase [Géron 2019]. Es decir, cada *fold* tendrá aproximadamente la misma proporción de instancias de cada clase que el conjunto de datos completo.

2.5 Trabajos relacionados

El problema de clasificación de reseñas puede abordarse a través de distintas técnicas de preprocesamiento y algoritmos de aprendizaje. Sin embargo, la decisión de qué combinación de técnicas utilizar se basa en el resultado de la métrica aplicada para medir el rendimiento del modelo generado. Por tal razón, se mencionan algunos trabajos relacionados con esta investigación clasificados según el tipo de métrica de evaluación empleada: combinadas o de rango.

2.5.1 Evaluación mediante Métricas Combinadas

En [Satriaji y Kusumaningrum 2018] realizaron un estudio sobre el impacto que tiene la técnica SMOTE al realizar balanceo de datos de tipo oversampling. La tarea llevada a cabo fue la de clasificar reseñas de hoteles obtenidas de la plataforma Traveloka mediante herramientas de web scraping. Recolectaron un total de 13,000 reseñas, de las cuales seleccionaron aleatoriamente 1,500 y las etiquetaron manualmente como positivas y negativas.

De igual forma aplicaron técnicas de preprocesamiento como la tokenización, eliminación de palabras vacías y algoritmos de normalización para corregir errores tipográficos. El uso de SMOTE se combinó con tres tipos de técnicas de extracción de características: Term Presence, Term Occurrence y TF-IDF. Los algoritmos de ML que utilizaron fueron: LR, NB y SVM.

El rendimiento de los modelos generados fue evaluado utilizando la métrica de tipo

combinada, G-Mean. Como resultado, obtuvieron que el uso de Term Occurrence en conjunto con LR tuvo un mejor rendimiento promedio, alcanzando un puntaje de 81.65 % de G-Mean. Los autores concluyeron que SMOTE es una técnica eficaz para mejorar el rendimiento de los modelos entrenados con conjuntos de datos desbalanceados.

Un año después, en [Dharma y Saragih 2022] realizaron una comparación de tres técnicas de extracción de características: BoW, TF-IDF e Improved TF-IDF para realizar la clasificación de reseñas del hotel Tabo Cottages, ubicado en Indonesia, las cuales fueron recuperadas de la plataforma Tripadvisor. Las reseñas fueron etiquetadas en positivas, negativas y neutras.

Como parte de su preprocesamiento, se balancearon las clases mediante oversampling para la clase minoritaria (negativa) y undersampling a las clases mayoritarias (positiva y neutra). A continuación, se realizó la eliminación de signos de puntuación y lematización. Como algoritmo de ML utilizaron SVM y evaluaron los resultados con métricas como la Exactitud, Precisión y Sensibilidad, así como el F-Score.

Como resultado, muestran que TF-IDF fue la técnica que produjo los mejores resultados, alcanzando un F-Score de 70.08 %. Los autores concluyeron que, según los resultados y características del conjunto de datos, la presencia de la misma palabra tiene una influencia significativa, siendo la razón por la que TF-IDF sobrepasó a las otras técnicas utilizadas.

2.5.2 Evaluación por Métricas de Rango

En [Gazali Mahmud et al. 2023] implementaron el algoritmo de kNN en conjunto con SMOTE, para abordar el problema del desbalance de clases al realizar un análisis de sentimientos aplicado a reseñas de hoteles. Se enfocaron en reseñas extraídas de tres hoteles populares en Labuan Bajo, con el objetivo de clasificar las reseñas como positivas o negativas y mejorar la precisión en la clasificación.

Como técnicas de extracción de características, hicieron uso de BoW, TF-IDF e Improved TF-IDF. El modelo fue entrenado con balanceo mediante SMOTE y sin él. En los resultados, se evaluó el rendimiento del modelo usando las métricas de Exactitud, Preci-

sión y Sensibilidad, pero sobre todo, como en trabajos anteriores y dada la naturaleza de los datos, hicieron uso de la métrica ROC-AUC.

Los resultados mostraron que, al aplicar SMOTE, el modelo mejoró significativamente el rendimiento en comparación con no utilizar ninguna técnica de balanceo. A través de ROC-AUC se alcanzaron valores que van del 82.1 % al 94.4 %, dependiendo de las reseñas del hotel utilizado para el entrenamiento. Además, identificaron factores clave que contribuyeron a la satisfacción de los turistas, como la limpieza, instalaciones y servicios.

2.6 Resumen

En este capítulo se abordaron los fundamentos teóricos y metodológicos relevantes para el desarrollo de esta tesis. Se inicia con una introducción al Procesamiento de Lenguaje Natural y su aplicación mediante el Análisis de Sentimientos, seguido de la descripción de técnicas de extracción de características que generan Modelos de Espacios Vectoriales para representar datos textuales en vectores numéricos, siendo estas técnicas las basadas en Bolsa de Palabras como TF-IDF y TF-IGM. Asimismo, se aborda el uso de la técnica estadística chi-cuadrado para realizar una selección de términos o características más representativas en dichos modelos generados.

Posteriormente, se aborda el problema del desbalance de clases en los datos mediante estrategias de submuestreo y sobremuestreo, que balancean la distribución de instancias en los datos con dos distintos enfoques, siendo el primero el balanceo en el espacio vectorial, con técnicas como Random Undersampling, Random Oversampling, SMOTE y DEBOHID; siendo el segundo, el aumento de datos basado en manipulación textual, con EDA.

En la sección de aprendizaje automático, se describe el funcionamiento de los algoritmos empleados para realizar la clasificación de la polaridad de reseñas. Estos son: Regresión Logística, que estima probabilidades mediante la optimización de la función logística; Clasificador Bayesiano Ingenuo, basado en la regla de Bayes y la independencia entre características; y Máquina de Soporte Vectorial, que optimiza un hiperplano para maximizar la separación entre clases.

Para evaluar el rendimiento de los modelos generados a partir de la combinación de diversas técnicas y algoritmos, se emplean distintas métricas. Entre ellas se incluyen la precisión, la sensibilidad, la especificidad, la métrica F1 y sus promedios macro, así como G-Mean y la exactitud balanceada. Estas métricas permiten analizar la efectividad del modelo desde múltiples perspectivas. Además, se describe el uso de la validación cruzada para obtener una estimación más robusta del desempeño.

Finalmente, se revisan trabajos relacionados que han utilizado métricas combinadas y de rango para la evaluación de modelos en tareas similares como ejemplos de aplicación de las técnicas y métricas expuestas.

3 Método Propuesto

Contenidos del Capítulo

3.1 Descripción General del Método 45
3.2 Etapas del Método Propuesto 48
3.3 Resumen 52

En este capítulo se presenta el método para clasificar la polaridad de reseñas en inglés sobre la experiencia de los clientes de servicios de hoteles todo incluido ubicados en Bahías de Huatulco, Oaxaca. El método propuesto parte del problema planteado, expresado con la siguiente pregunta:

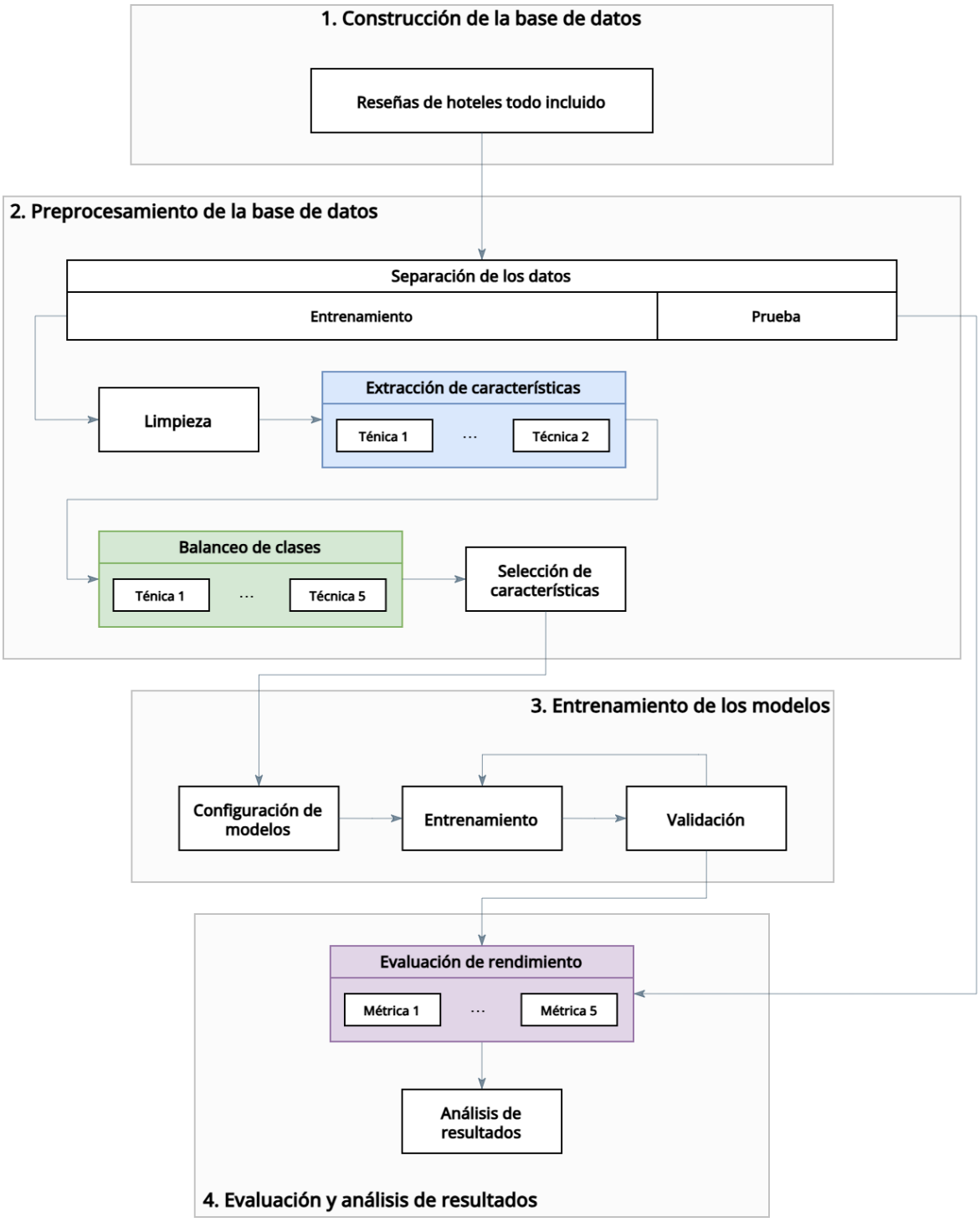
¿Qué técnicas de extracción de características y balanceo de clases, combinadas, pueden mejorar el desempeño de los algoritmos de aprendizaje automático en la clasificación de la polaridad positiva y negativa de reseñas en inglés sobre la experiencia de clientes en hoteles todo incluido?

El diseño del método parte de incluir una selección de la técnica de extracción de características adecuada; así como el uso de estrategias de balanceo de clases y la generación de modelos a partir de distintos algoritmos de aprendizaje automático que permitan maximizar el rendimiento de los modelos construidos.

3.1 Descripción General del Método

En esta sección, se presenta el método propuesto a partir del marco general del proceso genérico del análisis de sentimientos [Birjali et al. 2021]. Para esta investigación, se consideraron cinco principales etapas para el desarrollo del método propuesto; estas etapas se muestran en la Figura **3.1**.

Figura 3.1
Esquema del método propuesto para la investigación



En la primera etapa se realiza la construcción de la base de datos textuales e incluye la identificación y selección de la fuente de datos de reseñas de hoteles (ver Figura **3.1**, 1. Creación de la base de datos). También se realiza la extracción de los datos utilizando herramientas de *web scraping*, así como su organización y limpieza. Además, se lleva a cabo la asignación de etiquetas de polaridad a las reseñas en función de las puntuaciones, utilizando dos tipos de etiquetado: el Etiquetado A distingue entre negativas (1-2) y positivas (3-5), mientras que el Etiquetado B añade una clase neutra para las puntuaciones de 3. Finalmente, se realiza un análisis exploratorio para evaluar la calidad y diversidad de los datos recopilados.

En la segunda etapa se realiza el preprocesamiento de la base de datos textuales obtenida (ver Figura **3.1**, 2. Preprocesamiento de la base de datos). Esto abarca su separación en conjuntos de entrenamiento y prueba, para, posteriormente, realizar la limpieza y normalización del texto. También, se aplican técnicas para representar el texto en vectores numéricos: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) y Term Frequency-Inverse Gravity Moment (TF-IGM). Así como balanceo de datos, utilizando Random Oversampling, Random Undersampling, Synthetic Minority Oversampling Technique (SMOTE), Differential Evolution Based Oversampling approach for Highly Imbalanced Datasets (DEBOHID) e Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks (EDA). Finalmente, se seleccionan las características más relevantes mediante la técnica estadística de chi-cuadrado (χ^2).

En la tercera etapa se configuran y construyen los modelos predictivos para la clasificación de la polaridad (ver Figura **3.1**, 3. Entrenamiento de los modelos), donde los algoritmos: Regresión Logística (LR), el Clasificador Bayesiano Ingenuo o Naïve Bayes (NB) y la Máquina de Soporte Vectorial (SVM) en combinación con las distintas técnicas definidas en la segunda etapa, son entrenados y validados para estimar la robustez de los modelos generados.

En la última etapa se realiza la evaluación de los modelos obtenidos (ver Figura **3.1**, 4. Evaluación y análisis de resultados), por lo que se mide y analiza su rendimiento mediante métricas como: Precisión, Sensibilidad, Especificidad, Exactitud Balanceada, F-Score y G-Mean, con énfasis en estas dos últimas, que son las que validan o rechazan la hipótesis planteada.

3.2 Etapas del Método Propuesto

A continuación se detallan cada una de las etapas que conforman el método propuesto.

Etapa 1. Construcción de la base de datos

La primera etapa del método tiene como propósito construir una base de datos textuales de reseñas de hoteles todo incluido en Bahías de Huatulco, Oaxaca, al realizar las siguientes actividades:

1. **Selección de la fuente y recolección de datos.** Para seleccionar la fuente de donde se hará la extracción de datos, se siguen los siguientes pasos:
   (a) Identificar las fuentes de datos textuales, siendo estas las plataformas o sitios web que contienen reseñas de hoteles.
   (b) Seleccionar la fuente de la que se hará la extracción de datos.
   (c) Identificar los hoteles que cumplen con ser todo incluido y estar ubicados en Bahías de Huatulco, Oaxaca.
   (d) A partir de la información disponible en la fuente, se recopilarán los atributos presentes en cada reseña, por ejemplo, el contenido textual y las puntuaciones.
2. **Extracción de datos.** La extracción de datos de sitios web se realiza con una herramienta de *web scraping*. La extracción se dará mediante los siguientes pasos:
   (a) Implementar el script para recuperar los datos, adaptado a la estructura del documento HTML del sitio. El script debe ser ético y cumplir con las políticas de uso, evitando recopilar información personal.
   (b) Aplicar el script para recuperar los atributos seleccionados de las reseñas de cada hotel identificado.
3. **Organización y limpieza de los datos.** Una vez que los datos se hayan recopilado, es esencial realizar la limpieza y organización de los mismos, esto mediante los siguientes pasos:

(a) Integrar los datos recopilados de las reseñas por hotel en un único conjunto de datos.

(b) Eliminar caracteres problemáticos que puedan afectar la legibilidad de los textos, como los no imprimibles, corruptos o de codificación inconsistente.

(c) Eliminar las reseñas duplicadas para que, cada una de las reseñas en los datos sea única.

(d) Identificar datos vacíos y eliminarlos, ya que no aportan valor.

(e) Excluir atributos no relevantes para realizar la tarea de clasificación de polaridad de reseñas, es decir, mantener solo datos textuales y puntajes.

(f) Generar un nuevo atributo que integre los datos textuales seleccionados en el paso anterior en un único atributo consolidado.

4. **Etiquetado.** Asignar etiquetas de polaridad a las reseñas según la puntuación otorgada por el usuario. Siguiendo las propuestas de [Budhi et al. 2021], se emplean dos tipos de etiquetado:

   (a) Etiquetado A. Reseñas con puntuaciones de 1-2 se consideran negativas, mientras que las que van de 3-5 como positivas.

   (b) Etiquetado B. Reseñas con puntuaciones de 1-2 se consideran negativas, las de 3 como neutras y las de 4-5 como positivas.

5. **Análisis exploratorio.** El análisis exploratorio tiene como objetivo evaluar la calidad y diversidad de los datos recopilados. A continuación, se listan los tipos de análisis a realizar para esta actividad:

   (a) Visualización de la distribución de los datos por puntuaciones y etiquetados.

   (b) Conteo estadístico de las frecuencias por reseña en cuanto a la cantidad de caracteres y palabras, así como en la longitud promedio de las palabras y de las oraciones.

   (c) Representación visual de las tendencias en los datos: nube de palabras, y visualización de los datos en dos dimensiones.

Etapa 2. Preprocesamiento de la base de datos

El objetivo de la segunda etapa es preparar la base de datos para el entrenamiento de los modelos de aprendizaje automático en etapas posteriores, al llevar a cabo las siguientes actividades:

1. **Separación de los datos.** Dividir la base de datos en conjuntos de entrenamiento y prueba para evaluar la capacidad de generalización de los modelos. Los pasos son:
   (a) Dividir la base de datos en conjuntos de entrenamiento (80 %) y prueba (20 %) de forma estratificada para mantener la proporción de clases.
   (b) Verificar, mediante una contabilización, que cada conjunto refleje la distribución original de la base de datos.
2. **Limpieza de los datos textuales.** Se preparan los datos textuales mediante técnicas de limpieza y normalización, o **text wrangling**. Los pasos a seguir son:
   (a) Aplicar normalización. Lo que involucra convertir el texto a minúsculas, expandir contracciones, así como eliminar números, caracteres especiales y enlaces.
   (b) Aplicar tokenización para dividir los textos en palabras.
   (c) Eliminar palabras vacías (stopwords), excepto aquellas con connotación negativa, como “no”, “nor” y “not”, ya que eliminarlas puede influir en la interpretación del sentimiento expresado en una reseña.
   (d) Realizar lematización para reducir las palabras a su forma base.
3. **Extracción de características.** Implementar dos técnicas para representar el texto en vectores numéricos para la generación de los modelos finales. Estas técnicas generan modelos de espacios vectoriales (VSMs) que ponderan la frecuencia de los términos con diferentes enfoques, las cuales son:
   (a) TF-IDF, el enfoque no supervisado.
   (b) TF-IGM, el enfoque supervisado.

4. **Balanceo de clases.** Realizar un balanceo al conjunto de entrenamiento con el fin de equilibrar la cantidad de instancias presentes para cada clase, al implementar cinco técnicas de resampling. Dichas técnicas se agrupan según su forma de realizar el balanceo, las cuales son:

   (a) Random Oversampling, Random Undersampling, SMOTE y DEBOHID, que trabajan en el espacio vectorial.

   (b) EDA, que realiza un aumento de datos basado en manipulación textual.

5. **Selección de características.** Utilizar el método estadístico χ^2 para seleccionar un subconjunto de características óptimo que mantenga aquellos términos más relevantes en los VSMs generados por las técnicas de extracción de características con el fin de reducir su dimensionalidad.

Etapa 3. Entrenamiento de los modelos

Para la tercera etapa, el objetivo es entrenar y validar los modelos generados por la combinación de técnicas y algoritmos con los datos preparados y etiquetados, por lo que se realizan las siguientes actividades:

1. **Configuración de los modelos.** Se preparan los modelos de clasificación integrando las técnicas propuestas previamente. Los pasos son:

   (a) Definir los pipelines que combinen extracción de características y balanceo de clases.

   (b) Integrar los pipelines con los algoritmos de aprendizaje automático: LR, NB y SVM.

   (c) Configurar los parámetros iniciales de las técnicas y los algoritmos.

2. **Entrenamiento de los modelos.** Realizar el entrenamiento de cada uno de los modelos, utilizando los datos de entrenamiento preparados.

3. **Validación de los modelos.** Aplicar una validación cruzada de los modelos mediante *Stratified K-Fold Cross-Validation* con repetición para estimar el desempeño y robustez

de los modelos generados por cada combinación de técnicas y algoritmos. Para ello, se emplean $K = 10$ particiones y $R = 5$ repeticiones.

Etapa 4. Evaluación y análisis de resultados

En la cuarta y última etapa se tiene como fin evaluar el rendimiento de los modelos entrenados y analizar los resultados obtenidos para determinar el modelo más eficaz, efectuando las siguientes actividades:

1. **Evaluación del rendimiento de los modelos.** Evaluar el desempeño de los modelos entrenados, utilizando el conjunto de prueba previamente separado, mediante las métricas propuestas (Precisión, Sensibilidad, Especificidad, Exactitud Balanceada, F1 y G-Mean).
2. **Comparación de resultados.** Comparar los resultados obtenidos de la evaluación para identificar cuál de los modelos entrenados alcanza los mejores resultados en cuanto a la clasificación de polaridad de las reseñas de hoteles todo incluido en Bahías de Huatulco, Oaxaca.
3. **Análisis de los resultados.** Analizar los resultados obtenidos en la evaluación y la validación cruzada con la intención de establecer qué modelo es el más adecuado para la tarea de clasificación en este trabajo. Así también, determinar si el uso de las técnicas de preprocesamiento y métricas adecuadas aumentan el rendimiento de un modelo de aprendizaje automático.

3.3 Resumen

En este capítulo se describe el método propuesto, diseñado para validar o rechazar la hipótesis planteada a partir de la pregunta de investigación descrita en el Capítulo I de esta tesis. El método propuesto se describe inicialmente de manera general para mostrar una estructura en cuatro etapas principales. En la segunda sección se desglosa cada una de las etapas, a través de actividades.

En la primera etapa se define el procedimiento para la creación de la base de datos textuales conformada por reseñas de hoteles todo incluido con dos etiquetados de polaridad. En la segunda etapa se efectúa el preprocesamiento de dicha base de datos utilizando distintas técnicas de limpieza, extracción de características y balanceo de clases, con el objetivo de prepararla para los algoritmos de aprendizaje. Para la tercera etapa se generan y validan los modelos que realizan la clasificación de polaridad. Finalmente, en la cuarta etapa se realiza la evaluación de los modelos y análisis de los resultados obtenidos.

4 Resultados

Contenidos del Capítulo

4.1 Entorno de desarrollo y experimentación 56
4.2 Reseñas de hoteles todo incluido 57
4.2.1 Construcción de la base de datos 57
4.2.2 Análisis exploratorio de la base de datos 61
4.3 Preprocesamiento de la base de datos 69
4.3.1 Separación de los datos 69
4.3.2 Limpieza de los datos textuales 70
4.3.3 Extracción de características 71
4.3.4 Balanceo de clases 73
4.3.5 Selección de características 74
4.4 Experimento 1: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado A 76
4.4.1 Entrenamiento de modelos de clasificación 77
4.4.2 Evaluación del rendimiento de los modelos 86
4.4.3 Análisis y comparación de resultados 92
4.5 Experimento 2: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado B 95
4.5.1 Entrenamiento de modelos de clasificación 95
4.5.2 Evaluación del rendimiento de los modelos 104
4.5.3 Análisis y comparación de resultados 110

La identificación de polaridad de reseñas en inglés de hoteles todo incluido ubicados en Bahías de Huatulco, Oaxaca, mediante el uso de un enfoque supervisado, necesita de datos etiquetados, como las reseñas presentes en plataformas web, que cuenten con una puntuación asignada por el usuario.

En esta tesis se planteó como hipótesis que, el uso de una métrica de evaluación como la Media Geométrica o G-Mean, que considera por igual la recuperación de instancias positivas y negativas, puede ser más adecuada para evaluar este tipo de tareas que el F-Score o F1, ya que el principal enfoque de esta última es la clase de importancia. Esta métrica,

G-Mean, permitirá evaluar el impacto de la utilización de técnicas de extracción de características y balanceo de clases en modelos de aprendizaje automático para realizar esta identificación.

En este capítulo se presentan los resultados obtenidos mediante la aplicación del método propuesto, que plantea el uso en conjunto de técnicas de extracción de características sobre datos textuales con técnicas de balanceo de clases. Las técnicas de extracción de características utilizadas son: Term Frequency-Inverse Document Frequency (TF-IDF) que es una técnica no supervisada y Term Frequency-Inverse Gravity Moment (TF-IGM) que es supervisada. Las técnicas de balanceo de clases utilizadas fueron: Random Oversampling (ROS), Random Undersampling (RUS), Synthetic Minority Oversampling Technique (SMOTE), Differential Evolution Based Oversampling approach for Highly Imbalanced Datasets (DEBOHID) e Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks (EDA). Los algoritmos de aprendizaje utilizados para la experimentación son la Regresión Logística (LR), el Clasificador Bayesiano Ingenuo o Naïve Bayes (NB) y la Máquina de Soporte Vectorial (SVM). Estos algoritmos fueron seleccionados de acuerdo a los implementados en los trabajos relacionados en el estado del arte.

4.1 Entorno de desarrollo y experimentación

Para el desarrollo y realización de los experimentos se hizo uso de un equipo de cómputo con los componentes descritos en la Tabla **4.1**.

Tabla 4.1
Especificaciones técnicas del equipo de cómputo utilizado

Componente	Especificación
Marca	Asus TUF A14 2024
Sistema operativo	Windows 11 con Windows Subsystem for Linux (WSL)
Procesador	AMD Ryzen AI 9 HX 370
Memoria RAM	16 GB
Tarjeta gráfica	NVIDIA GeForce RTX 4060

La implementación y prueba de los modelos se realizó utilizando el lenguaje de programación Python versión 3.12.0, con las bibliotecas mencionadas en la Tabla **4.2**.

Tabla 4.2
Bibliotecas de Python utilizadas para la implementación

Biblioteca	Versión
NumPy	2.1.2
NLTK	3.9.1
Scikit-Learn	1.5.2
Imbalanced-Learn	0.12.4
Pandas	2.2.3

La visualización de los resultados se hizo mediante las bibliotecas Matplotlib versión 3.9.2 y Seaborn versión 0.13.2. Así mismo, el editor de código utilizado fue Visual Studio Code conectado a WSL, por el cual se ejecutaron los experimentos.

4.2 Reseñas de hoteles todo incluido

En esta sección se presentan los resultados de la construcción de la base de datos, que incluyó la identificación y selección de la fuente de datos, la extracción de reseñas mediante *web scraping*, así como su posterior limpieza, organización y etiquetado. Así mismo, se muestra un análisis exploratorio de la base de datos resultante.

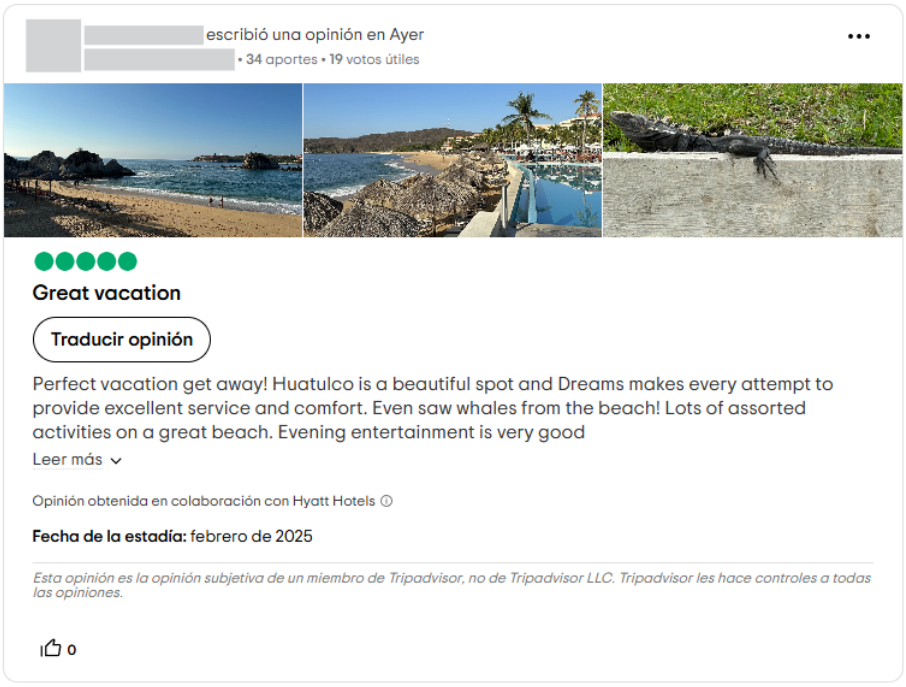
4.2.1 Construcción de la base de datos

El primer paso para la construcción de la base de datos partió de la **identificación de la fuente de información**. Después de realizar una búsqueda en Internet, las principales plataformas web que proveen información acerca de hoteles todo incluido ubicados en Bahías de Huatulco, Oaxaca, son Tripadvisor y Booking. En total, se identificaron cinco hoteles todo incluido en esta bahía: Park Royal Beach Resort Huatulco, Las Brisas Huatulco,

Barceló Huatulco, Secrets Huatulco Resort & Spa y Dreams Huatulco Resort & Spa.

Se seleccionó la plataforma de reseñas más utilizada en el estado del arte y la que ofrecía la mayor cantidad de datos sobre reseñas de hoteles en Bahías de Huatulco, Oaxaca: Tripadvisor (https://www.tripadvisor.com) Otra razón para esta selección es que, cada reseña tiene un puntaje en una escala que va de 1 a 5, acorde para el etiquetado en el método propuesto.

Figura 4.1
Ejemplo de una reseña para el hotel Dreams Huatulco Resort & Spa



Nota. Reseña pública recuperada de la plataforma Tripadvisor sobre el hotel *Dreams Huatulco Resort & Spa*. (Reseña original disponible en: https://www.tripadvisor.com).

Los datos personales, como el nombre y foto de perfil del usuario, así como su lugar de procedencia, se consideran datos sensibles y, por lo tanto, no fueron utilizados. De cada una de las reseñas disponibles en Tripadvisor (ver ejemplo en Figura 4.1), se recopilaron los siguientes atributos:

1. Nombre del hotel.
2. Título de la reseña.
3. Texto de la reseña.
4. Fecha de la estadía.
5. Puntuación otorgada por el usuario.

El **proceso de extracción** de información se realizó en cada uno de los hoteles, mediante la implementación de un script que inspecciona el código HTML utilizando JavaScript. Dicho script obtiene los atributos determinados para cada reseña utilizando selectores CSS y los va almacenando en un objeto de JavaScript (ver la Figura 4.2) que posteriormente es guardado en un archivo JSON. Al final, se obtuvieron cinco archivos JSON con los datos en crudo, recuperándose un total de **19,369** reseñas.

Figura 4.2
Ejemplo de una reseña extraída para el hotel Barceló Huatulco


```
{
  "hotel_name": "Barceló Huatulco",
  "title": "family vacations",
  "date": "June 2024",
  "score": 3,
  "text": "I have been in this same hotel Barcelo
    but in other destinations and they have all
    been wonderful, Barcelo Huatulco is good, but
    I think it is a little overrated."
}
```


Como parte de la **organización y limpieza de los datos**, todos los registros se integraron en un solo conjunto de datos con los cinco archivos JSON. Al conjunto de datos resultante se realizó una eliminación de caracteres problemáticos, como por ejemplo, emojis y símbolos especiales, entre otros. En total se eliminaron **23,301** caracteres, como se muestra en la Tabla 4.3.

Tabla 4.3
Conteo de caracteres problemáticos por atributo

Atributo	Caracteres
Nombre del hotel	3461
Título de la reseña	447
Texto de la reseña	19393
Fecha de la estadía	0

A continuación, se eliminaron las reseñas duplicadas. En total, se identificaron **84** duplicados que presentaban exactamente la misma información en todos los atributos, por lo que fueron eliminadas. Un ejemplo de reseñas duplicadas se muestra en la Figura **4.3**.

Figura 4.3
Ejemplo de reseñas duplicadas en los datos

Dreams Huatulco Resort & Spa	1 week of total relaxation and fun	March 2015	5	By far the best all inclusive ever. Food was d...
Dreams Huatulco Resort & Spa	1 week of total relaxation and fun	March 2015	5	By far the best all inclusive ever. Food was d...
Las Brisas Huatulco	100% recommended I would return a thousand times	January 2024	5	All the food was delicious, the drinks were we...
Las Brisas Huatulco	100% recommended I would return a thousand times	January 2024	5	All the food was delicious, the drinks were we...

En este sentido, se verificó la presencia de reseñas con datos faltantes o vacíos, detectándose **seis** casos. **Cinco** de ellas carecían de título, mientras que **una** no contenía el texto de la reseña. Se determinó que no era necesario eliminar las reseñas sin título, ya que este atributo solo complementa el contenido principal. Sin embargo, la reseña sin texto fue eliminada, puesto que el texto es el atributo principal para realizar la clasificación.

Posteriormente, se realizó una exclusión de atributos con información irrelevante para realizar la clasificación de polaridad. Se consideró que el nombre del hotel y la fecha de estadía no aportan información significativa para la tarea, ya que el primero no es más que una manera de identificar a qué hotel pertenece la reseña, mientras que la fecha, de igual forma, al ser un dato temporal, no es un factor determinante para saber la polaridad de una

reseña; por lo que fueron descartados.

Con el fin de tener un solo atributo textual, se combinaron el título y el texto de la reseña, ya que esto permite una representación más completa de la opinión del turista, preservando información relevante para la clasificación de polaridad. Finalmente, el conjunto de datos fue **etiquetado** en función de la puntuación otorgada por el usuario para los dos tipos de etiquetados propuestos.

La base de datos construida se nombró **HuatulcoResortReviews**: La palabra «Huatulco» se refiere a la ubicación; «Resort» hace referencia a los complejos turísticos de los que se obtuvieron los datos, lo que también resalta la categoría premium de estos complejos; y «Reviews» indica que la base de datos recopila opiniones de clientes.

HuatulcoResortReviews está compuesta por un total de **19,284** reseñas, donde, para el Etiquetado A, **18,046** se consideran positivas y **1,238** como negativas, mientras que para el Etiquetado B, **16,064** se consideran positivas, **1,982** como neutras y **1,238** como negativas. En la Tabla 4.4 se observa una descripción del contenido de cada columna en la base de datos.

Tabla 4.4
Descripción de las columnas de la base de datos HuatulcoResortReviews

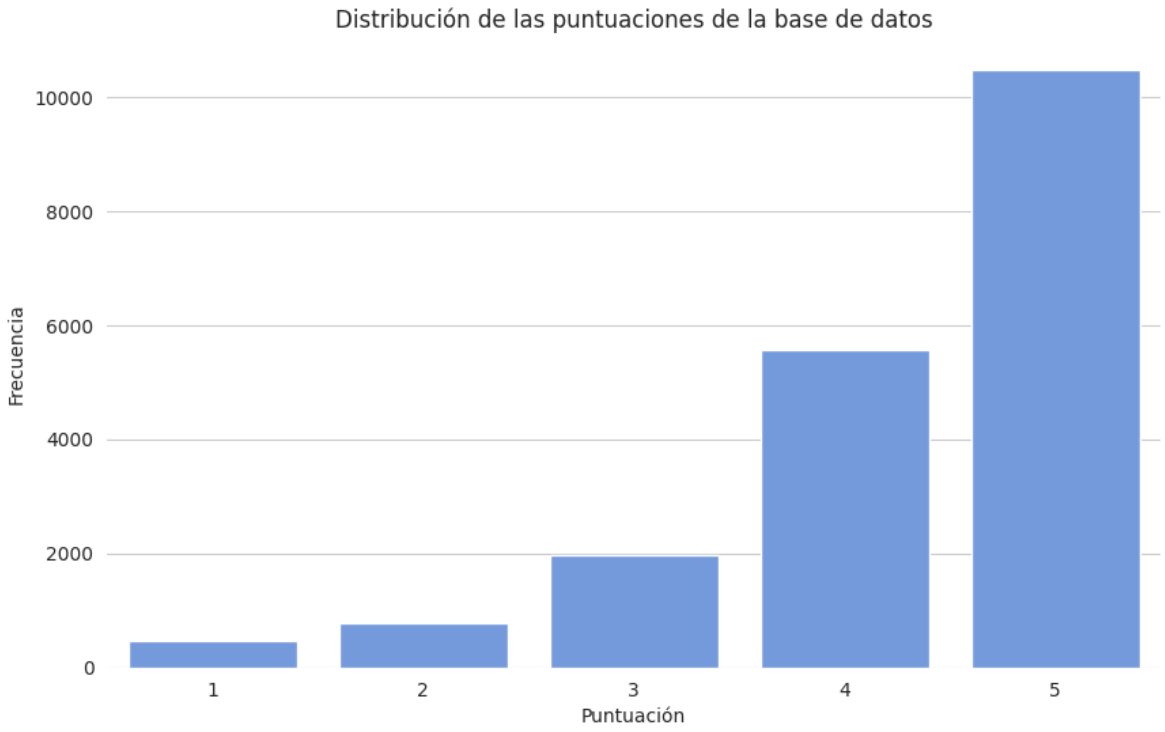
Columna	Descripción
review	Texto de la reseña del cliente sobre su experiencia en el hotel
score	Calificación numérica otorgada por el cliente, que representa su nivel de satisfacción (1-5)
a_label	Etiqueta correspondiente al Etiquetado A
b_label	Etiqueta correspondiente al Etiquetado B

4.2.2 Análisis exploratorio de la base de datos

En la Figura 4.4 se presenta la distribución de puntuaciones de las reseñas de la base de datos **HuatulcoResortReviews**. A partir de la muestra analizada, se observa que, conforme la puntuación aumenta, también lo hace la cantidad de reseñas. Esto podría indicar

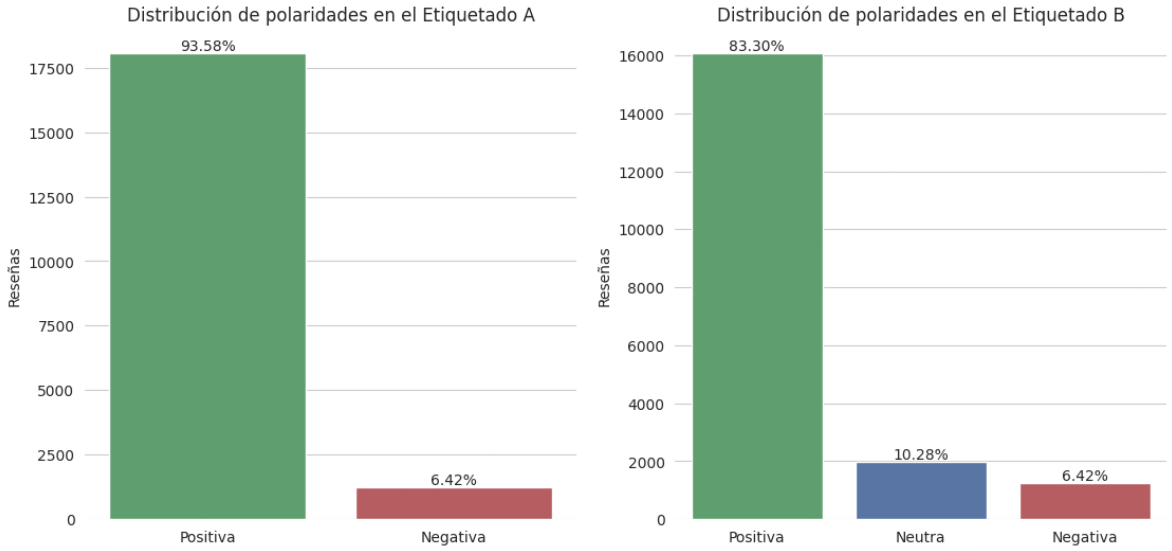
una mayor proporción de turistas con experiencias favorables, lo cual se refleja tanto en la calificación como en el número de reseñas registradas.

Figura 4.4
Distribución de puntuaciones en la base de datos HuatulcoResortReviews



En la Figura 4.5 se observa la distribución de las polaridades de acuerdo al tipo de etiquetado (Etiquetado A y Etiquetado B). En ambos tipos de etiquetado, la clase positiva tiene un 93.58 % y un 83.30 % del total de reseñas. Mientras que la clase negativa posee solo el 6.42 % de reseñas en ambas, denotando un gran porcentaje de diferencia en comparación con la positiva. Lo mismo ocurre con la polaridad neutra en el Etiquetado B, que abarca el 10.28 % del total de reseñas. Esto demuestra que hay una alta aceptación de los servicios y que la clase positiva es la mayoritaria.

Figura 4.5
Distribución de polaridades para ambos etiquetados en HuatulcoResortReviews



El Etiquetado A en **HuatulcoResortReviews** muestra una relación de desbalance de **1:14** ; es decir, por cada reseña negativa existen alrededor de 14 reseñas positivas. Por otro lado, para el Etiquetado B muestra una relación de desbalance de **1:8** entre la polaridad positiva y neutra, y de **1:12** entre la polaridad positiva y negativa.

En la Figura 4.6 y la Figura 4.7 se muestran cuatro histogramas con el conteo estadístico de las frecuencias por reseña, de la cantidad de caracteres y palabras, así como en la longitud promedio de las palabras y de las oraciones separadas por clase para ambos etiquetados. Se puede observar que, para todas las polaridades, existen las mismas tendencias: reseñas con una cantidad de caracteres menor a 3000, con menos de 600 palabras, y estas palabras cuentan con una longitud promedio entre 4 y 6 caracteres, así como una longitud promedio de las oraciones en palabras menor a 200. De acuerdo a estos valores, este tipo de información no resulta útil para mejorar la separación entre clases, por lo que no se usarán como parte de las características.

Figura 4.6
Frecuencias de ocurrencias de caracteres y palabras en Etiquetado A en HuatulcoResortReviews

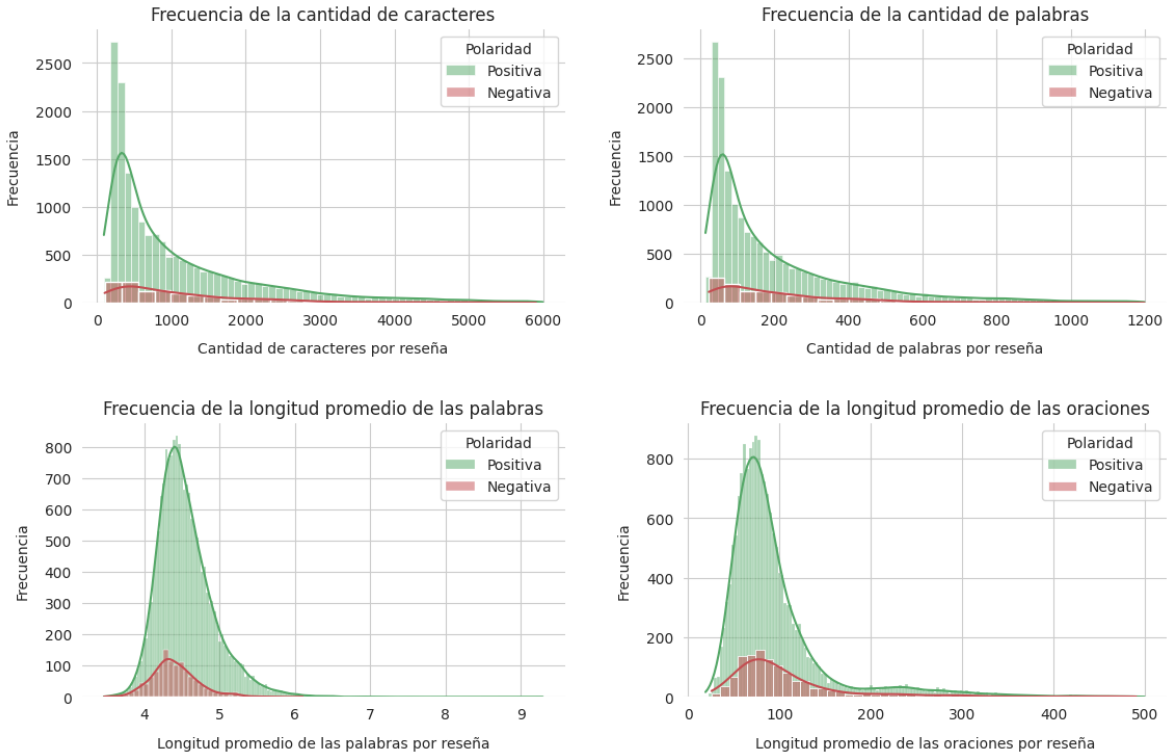
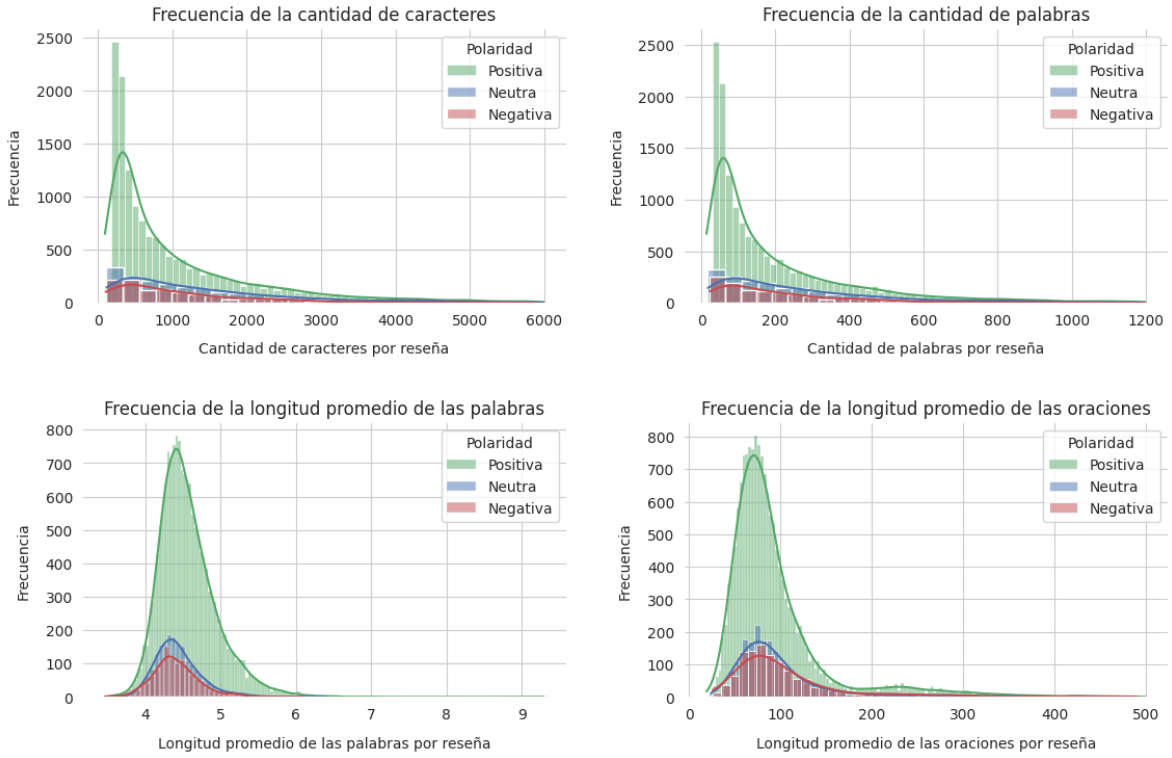


Figura 4.7
Frecuencias de ocurrencias de caracteres y palabras en Etiquetado B en HuatulcoResortReviews



La figura 4.8 muestra una nube de palabras en unigramas, bigramas y trigramas para el Etiquetado A de **HuatulcoResortReviews**, divididas por polaridad. Como se puede observar, en unigramas destacan términos como «room», «food», «resort», «restaurant» y «service» que reflejan el enfoque en instalaciones y servicios, pero no dan mucha más información.

A partir de bigramas se empieza a notar la diferencia entre polaridades, ya que se muestran frases como «go back», «food good» y «staff friendly» para las reseñas positivas, así como «could not», «would not» y «not good» para las negativas. Sin embargo, todavía no ofrecen un contexto lo suficientemente amplio para captar la idea que se pretende transmitir.

Es en los trigramas que se acentúa más la diferencia entre la polaridad positiva y nega-

tiva, ya que se observan frases como «would go back» y «would highly recommend» para las reseñas positivas, así como «never go back» y «would not recommend» para las negativas que reflejan de manera más clara la intención y sentimiento expresados por el cliente.

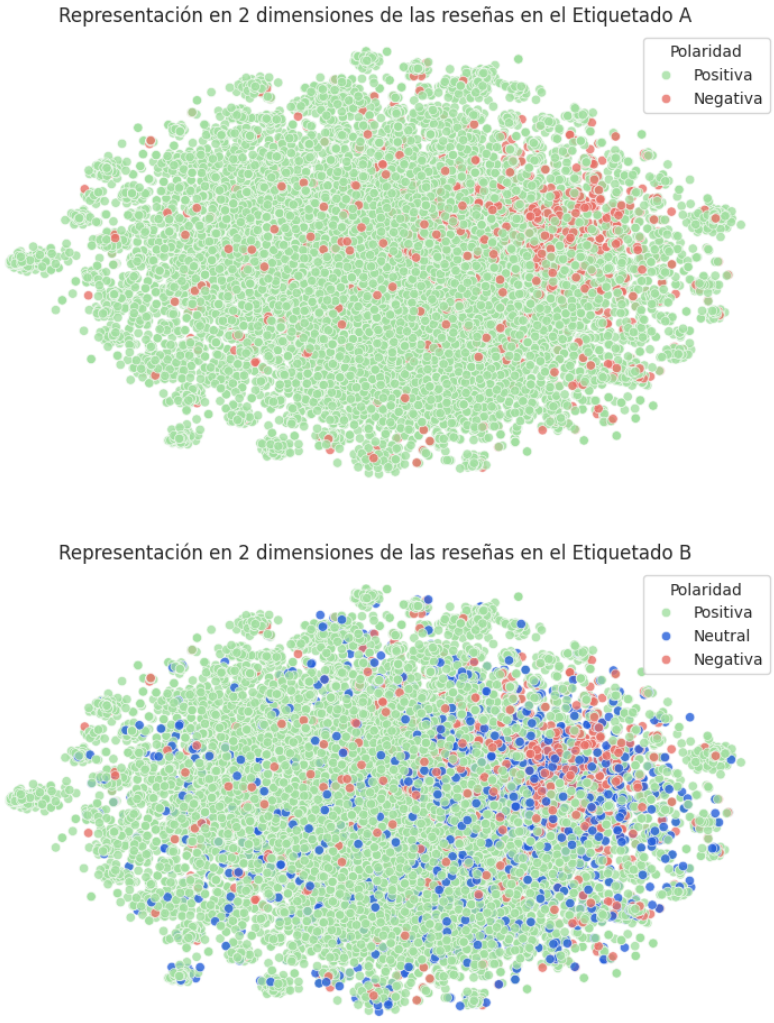
Figura 4.8
Nubes de n-gramas por polaridad en Etiquetado A en HuatulcoResortReviews

En la Figura 4.9 se muestran nubes de palabras en unigramas, bigramas y trigramas, ahora para el Etiquetado B de **HuatulcoResortReviews**. Para este etiquetado se presenta la misma situación vista en el Etiquetado A para las polaridades positiva y negativa. Sin embargo, la diferencia viene en la polaridad neutra, que tiende a presentar una combinación de términos o frases que pueden ser positivas o negativas. Algunos ejemplos son: «room clean» y «not good» en bigramas, así como «not go back» y «would go back» en trigramas.

Figura 4.9
Nubes de n-gramas por polaridad en Etiquetado B en HuatulcoResortReviews

Finalmente, en la Figura **4.10** se presentan las características por clase que muestran la distribución de las reseñas en un espacio vectorial de dos dimensiones, donde cada círculo representa una reseña. La reducción a dos dimensiones se realizó mediante la técnica t-SNE, que busca mantener las instancias similares cercanas y las distintas separadas. Esta técnica se utiliza principalmente para visualizar agrupaciones de instancias en un espacio de alta dimensionalidad, como los construidos con técnicas de vectorización de textos [Géron 2019].

Figura 4.10
Representación en dos dimensiones de las reseñas en HuatulcoResortReviews mediante t-SNE



4.3 Preprocesamiento de la base de datos

En esta sección se presenta el resultado de la ejecución del preprocesamiento en la base de datos nombrada **HuatulcoResortReviews**. El proceso comienza con la separación de los datos en los conjuntos: entrenamiento y prueba. Para después realizar los pasos de limpieza, extracción y selección de características, así como el balanceo de clases.

4.3.1 Separación de los datos

La base de datos **HuatulcoResortReviews** fue separada para el entrenamiento y evaluación de predicción de los modelos generados en las siguientes secciones. La separación de la base de datos se realizó en una proporción 80-20, para datos de entrenamiento y prueba respectivamente. Dado que en el análisis exploratorio de datos se detectó un alto desbalance en la base de datos **HuatulcoResortReviews**, la separación se realizó de forma estratificada para preservar la proporción original de instancias entre clases.

Tabla 4.5
Distribución de clases en los conjuntos de entrenamiento y prueba para Etiquetado A

Clase	HuatulcoResortReviews-A-Train	HuatulcoResortReviews-A-Test
Positiva	14437	3609
Negativa	990	248
Total	15427	3857

Nota. Se muestra la distribución de ejemplos pertenecientes a las clases positiva y negativa en los conjuntos de entrenamiento y prueba correspondientes al Etiquetado A.

Tabla 4.6
Distribución de clases en los conjuntos de entrenamiento y prueba para Etiquetado B

Clase	HuatulcoResortReviews-B-Train	HuatulcoResortReviews-BTest
Positiva	12851	3213
Neutra	1586	396
Negativa	990	248
Total	15427	3857

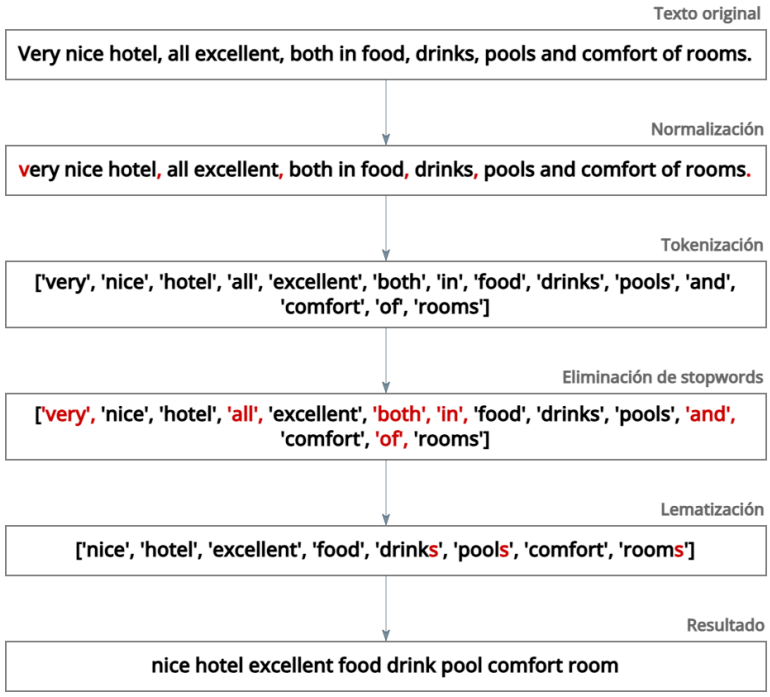
Nota. Se muestra la distribución de ejemplos pertenecientes a las clases positiva, neutra y negativa en los conjuntos de entrenamiento y prueba correspondientes al Etiquetado B.

En la Tabla 4.5 y la Tabla 4.6 se muestra la distribución de los datos en los conjuntos generados. En ambos etiquetados, se tiene un total de **15,427** instancias para entrenamiento y **3,857** para prueba. Similar a la base de datos original, los conjuntos **HuatulcoResort-Reviews-A-Train** y **HuatulcoResortReviews-A-Test** tienen una relación de desbalance de **1:14**, mientras que **HuatulcoResortReviews-B-Train** y **HuatulcoResortReviews-B-Test** presentan una relación de **1:8** y de **1:12**.

4.3.2 Limpieza de los datos textuales

Para cada uno de los textos en la columna *review* de los conjuntos generados en la Sección 4.3.1, se aplicaron técnicas de limpieza y normalización, o *text wrangling*, como el ejemplo que se observa en la Figura 4.11. El proceso empieza por normalizar el texto, en este caso convirtiendo a minúsculas y eliminando caracteres especiales. Lo siguiente es realizar la tokenización del texto para eliminar palabras vacías y lematizarlas para reducirlas a su forma base.

Figura 4.11
Proceso de limpieza para los datos textuales



El resultado de aplicar este proceso son textos limpios, formados únicamente por términos representativos de las ideas transmitidas por quien los escribió, lo que reduce el ruido y la dimensionalidad en etapas posteriores.

4.3.3 Extracción de características

Con los textos preparados, se realizó la **extracción de características**. Para TF-IDF se usó la implementación de scikit-learn: **TfidfVectorizer**. Para la ejecución se utilizaron sus parámetros por defecto, excepto los siguientes: $min\_df = 20$ y $max\_df = 0.9$. Estos parámetros ajustan los umbrales de frecuencia mínima y máxima de aparición de un término o palabra, con un mínimo de 20 documentos y un máximo del 90 % del total, con el fin de filtrar términos demasiado raros o demasiado comunes. Aplicando la técnica sobre los conjuntos **HuatulcoResortReviews-A-Train** y **HuatulcoResortReviews-B-Train**, se generaron vocabularios con un total de **5,527** y **5,522** términos o características, respecti-

vamente.

Por otro lado, en cuanto a TF-IGM, al ser una técnica de ponderación menos conocida que TF-IDF, no existe una implementación disponible en las bibliotecas de Python para Machine Learning. Por ello, fue necesario desarrollar una implementación propia basada en el trabajo de [Chen et al. 2016]. Se creó la clase **TfigmVectorizer**, que hereda de **CountVectorizer**, la clase base de **TfidfVectorizer**, con el objetivo de mantener la interoperabilidad con otras herramientas de la biblioteca.

Para verificar que la implementación sea correcta, se utilizaron los siguientes ejemplos que proporciona el artículo para el cálculo del factor IGM:

Ejemplo 1:

- Frecuencia de t_1 en cada clase: $[4, 2, 2, 2, 2]$
- Frecuencia de t_2 en cada clase: $[4, 8, 0, 0, 0]$
- Valor del parámetro λ: 7
- Resultados esperados: $IGM(t_1) = 1.875$, $IGM(t_2) = 4.5$

Ejemplo 2:

- Frecuencia de t_1 en cada clase: $[8, 7, 6, 6, 0, 0]$
- Frecuencia de t_2 en cada clase: $[9, 2, 2, 2, 0, 0]$
- Valor del parámetro λ: 7
- Resultados esperados: $IGM(t_1) = 1.875$, $IGM(t_2) = 3.333$

Para cada ejemplo se considera que existen 5 clases, por lo mismo, cada término cuenta con 5 conteos de frecuencia para cada una de las clases. La Figura **4.12** muestra que la

implementación del factor IGM genera los resultados esperados en ambos casos, validando su correcto funcionamiento.

Figura 4.12
Ejemplos del cálculo del factor IGM

```
------------------------------------------------------------

Factores IGM para ejemplo 1: [1.875 4.5  ]

Factores IGM para ejemplo 2: [1.875      3.33333333]

------------------------------------------------------------
```

Al igual que en **TfidfVectorizer**, en **TfigmVectorizer** se establecieron los parámetros $min\_df = 20$ y $max\_df = 0.9$. Los vocabularios generados para los conjuntos **HuatulcoResortReviews-A-Train** y **HuatulcoResortReviews-B-Train** son los mismos que los obtenidos con **TfidfVectorizer**, ya que ambos basan su implementación en la Bolsa de Palabras (BoW) para su construcción.

4.3.4 Balanceo de clases

Con las características preparadas, y debido a la existencia de un alto desbalance entre las clases de polaridad en ambos etiquetados (ver Figura 4.5), se aplicaron técnicas de **balanceo de clases**. Para las técnicas ROS, RUS y SMOTE, se utilizaron las implementaciones en la biblioteca imbalanced-learn. Tanto para ROS como para RUS se utilizaron sus parámetros por defecto, mientras que para SMOTE se estableció $k\_neighbors = 211$.

La técnica DEBOHID, al ser reciente y poco conocida, no cuenta con implementación en bibliotecas conocidas. Por ello, se desarrolló como una función en Python siguiendo el algoritmo propuesto por [Kaya et al. 2021]. Para aplicar el balanceo, se establecieron los siguientes parámetros para la técnica: $scale\_factor = 2$ y $crossover\_rate = 0.2$.

Por otro lado, la técnica EDA tampoco dispone de implementación en las bibliotecas, pero sus autores publicaron el código fuente en GitHub[1], lo que permitió usarlo directa-

[1] Disponible en: https://github.com/jasonwei20/eda_nlp

mente sin necesidad de reimplementarlo. En este caso, se emplearon los parámetros predeterminados.

4.3.5 Selección de características

Antes de entrenar los modelos, y una vez balanceados los conjuntos, es necesario determinar la cantidad de características a utilizar mediante un proceso de selección. Esto es fundamental al utilizar características generadas mediante técnicas basadas en BoW, ya que sufren de la conocida *maldición de la alta dimensionalidad*. Este fenómeno se refiere a la tendencia de los conjuntos de datos con un gran número de atributos o características a volverse dispersos, lo que dificulta el aprendizaje, incrementa el riesgo de sobreajuste y reduce la fiabilidad de las predicciones [Géron 2019].

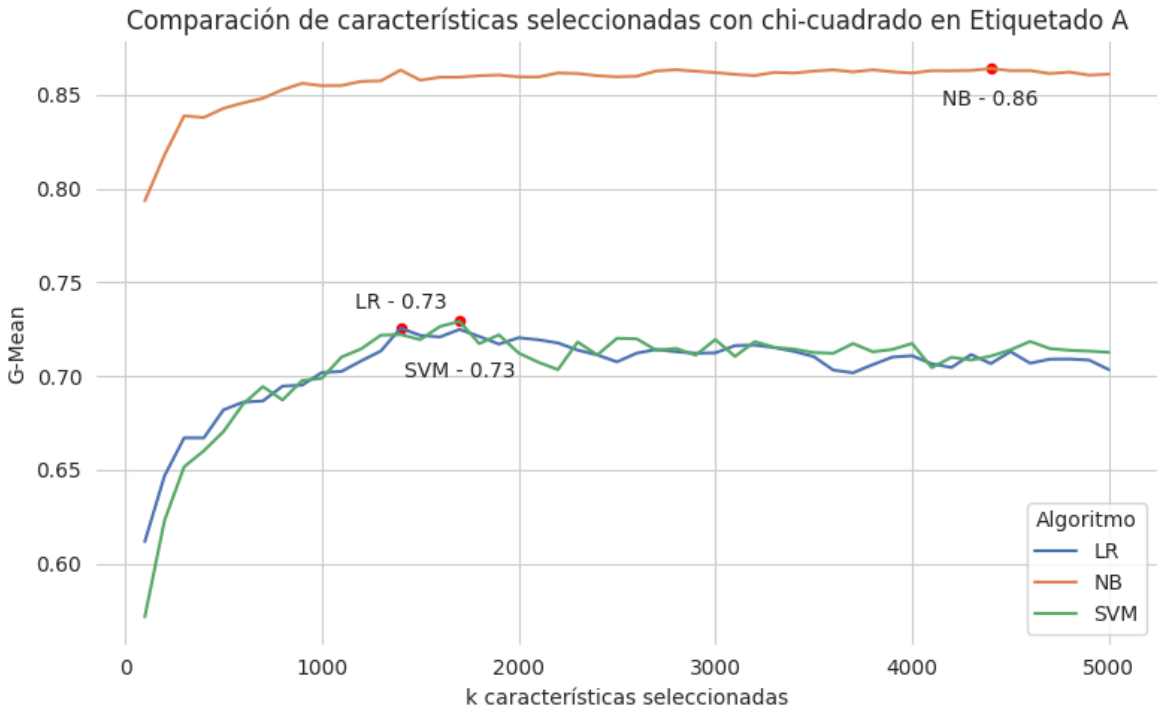
Se utilizó la técnica estadística chi-cuadrado (χ^2) para seleccionar el mejor subconjunto de k características que mantenga o pueda mejorar el rendimiento de los modelos. Para determinar el valor de k, se utilizó BoW para vectorizar los textos, ya que ofrece una representación neutral sin ponderar los términos. Así mismo, se emplearon los parámetros definidos para las técnicas TF-IDF y TF-IGM en la Sección **4.3.3** ($min\_df = 20$ y $max\_df = 0{,}9$). Además de eso, no se aplicó ningún tipo de preprocesamiento adicional sobre los textos, ni ajuste de hiperparámetros en los algoritmos.

Se llevó a cabo una validación cruzada estratificada con K-Fold, usando 10 particiones sobre los conjuntos **HuatulcoResortReviews-A-Train** y **HuatulcoResortReviews-B-Train**. Los subconjuntos de características seleccionados comenzaron en $k = 100$ y aumentaron en pasos de 100 hasta llegar a 5,000, considerando este último como un límite adecuado, dado que el número total de características generadas por las técnicas de extracción en cada etiquetado (ver Sección **4.3.3**) se aproxima a este valor.

En la Figura **4.13** se presentan los resultados al seleccionar diferentes valores de k para el Etiquetado A. Como se puede observar, para LR y SVM, a partir de las 1,000 características, el desempeño de los tres algoritmos permanece prácticamente igual, variando entre 0.01 y 0.03, siendo el valor máximo 0.73, alcanzado con aproximadamente 1,500 características. En el caso de NB, el desempeño se mantiene entre 0.85 y 0.86. De acuerdo a

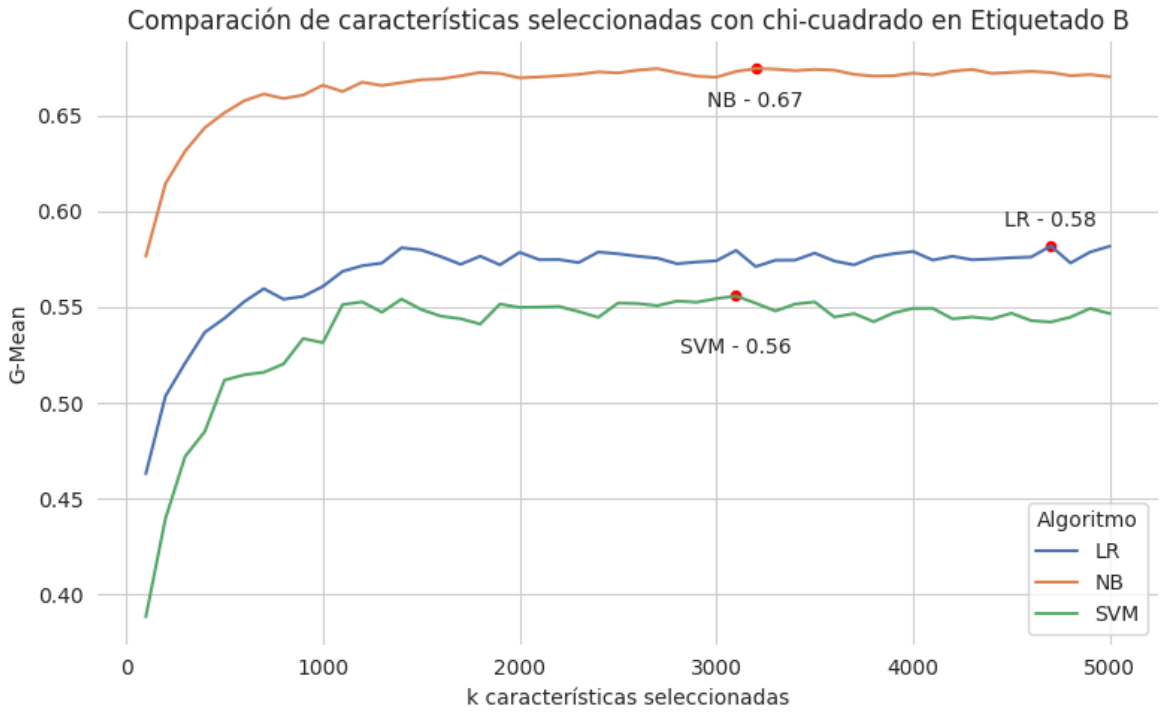
estos resultados, a partir de esta cantidad el rendimiento se mantiene constante, por lo que se seleccionaron **1,500** características para la construcción de los modelos.

Figura 4.13
Resultados de la selección de las mejores k características en Etiquetado A



En la Figura 4.14 se muestran los resultados obtenidos para el Etiquetado B. Se observa que, a partir de las 1,500 características, el desempeño de los tres algoritmos se mantiene sin cambios, manteniendo valores que varían entre 0.01 y 0.02. Por lo tanto, al igual que en el primer etiquetado, se seleccionarán **1,500** características.

Figura 4.14
Resultados de la selección de las mejores k características en Etiquetado B



4.4 Experimento 1: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado A

Después de haber especificado el preprocesamiento para la base de datos **HuatulcoResortReviews** con Etiquetado A, en esta sección se muestra la integración de la Sección 4.3 para el entrenamiento de modelos a partir de los siguientes algoritmos:

- Regresión Logística (LR)
- Clasificador Bayesiano Ingenuo (NB) y
- Máquina de Soporte Vectorial (SVM)

De acuerdo con los pasos especificados en la Etapa 3 del método propuesto en el capítulo 3 de esta tesis, sobre el entrenamiento y validación de los modelos. En la primera sección, se muestran los modelos resultantes de una validación cruzada mediante K-Fold estratificado (SKCV, por sus siglas en inglés de *Stratified K-Fold Cross-Validation*). En total, se obtuvieron treinta modelos que combinan las técnicas de vectorización o extracción de características TF-IDF y TF-IGM, junto con las técnicas de balanceo de clases: ROS, RUS, SMOTE, DEBOHID y EDA. Posteriormente, se realizó la evaluación y análisis de los resultados, de acuerdo a la Etapa 4 del método propuesto, utilizando las métricas: Exactitud Balanceada (BA, por sus siglas en inglés de *Balanced Accuracy*), G-Mean y F1-Score.

4.4.1 Entrenamiento de modelos de clasificación

En la etapa de entrenamiento, se utilizó el conjunto *HuatulcoResortReviews-A-Train* para el entrenamiento, que se implementó con la biblioteca scikit-learn de acuerdo a las especificaciones de la Sección **4.1**. A continuación, se listan los algoritmos con sus respectivos parámetros:

- **LogisticRegression** (LR): $penalty = l1$, $C = 0.1$, $solver = liblinear$, $max\_iter = 1000$.
- **LinearSVC** (SVM): $penalty = l1$, $C = 0.1$, $dual = False$, $max\_iter = 1000$.
- **MultinomialNB** (NB): con los parámetros por defecto.

La Tabla **4.7** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **LR** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el intervalo de confianza (CI) al 95 %. El modelo con mejor desempeño fue el que utilizó ROS como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 87.59 (± 0.37), una BA de 87.61 (± 0.37) y un F1 de 70.55 (± 0.33).

Tabla 4.7
Validación cruzada de los modelos que utilizan TF-IDF y LR para Etiquetado A

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
ROS	89.19 ± 0.06	71.36 ± 0.07	89.18 ± 0.06	87.61 ± 0.37	70.55 ± 0.33	87.59 ± 0.37
DEBOHID	89.12 ± 0.06	71.49 ± 0.06	89.12 ± 0.06	87.59 ± 0.44	70.64 ± 0.37	87.58 ± 0.44
EDA	88.73 ± 0.06	70.42 ± 0.06	88.72 ± 0.06	87.37 ± 0.44	69.65 ± 0.32	87.36 ± 0.44
SMOTE	86.87 ± 0.08	71.60 ± 0.08	86.84 ± 0.08	85.60 ± 0.44	70.87 ± 0.39	85.51 ± 0.45
RUS	74.26 ± 0.22	55.33 ± 0.17	74.24 ± 0.22	74.15 ± 0.56	55.28 ± 0.34	74.12 ± 0.56

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.8 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **NB** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó RUS como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 89.03 (± 0.39), una BA de 89.09 (± 0.40) y un F1 de 69.48 (± 0.33).

Tabla 4.8
Validación cruzada de los modelos que utilizan TF-IDF y NB para Etiquetado A

Balanceo	Entrenamiento			Prueba		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
RUS	91.11 ± 0.07	70.48 ± 0.18	90.98 ± 0.07	89.09 ± 0.40	69.48 ± 0.33	89.03 ± 0.39
SMOTE	91.51 ± 0.06	72.13 ± 0.07	91.43 ± 0.06	88.94 ± 0.44	70.60 ± 0.35	88.91 ± 0.43
ROS	92.18 ± 0.04	73.73 ± 0.05	92.12 ± 0.04	88.85 ± 0.38	71.78 ± 0.29	88.84 ± 0.38
DEBOHID	92.08 ± 0.06	74.36 ± 0.07	92.04 ± 0.06	88.65 ± 0.42	72.38 ± 0.29	88.64 ± 0.42
EDA	92.44 ± 0.06	76.45 ± 0.05	92.43 ± 0.06	88.14 ± 0.48	73.85 ± 0.35	88.09 ± 0.49

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla **4.9** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **SVM** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 88.65 (± 0.44), una BA de 88.68 (± 0.43) y un F1 de 73.86 (± 0.33).

Tabla 4.9
Validación cruzada de los modelos que utilizan TF-IDF y SVM para Etiquetado A

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
EDA	92.61 ± 0.06	76.36 ± 0.06	92.59 ± 0.06	88.68 ± 0.43	73.86 ± 0.33	88.65 ± 0.44
DEBOHID	92.54 ± 0.07	76.30 ± 0.08	92.53 ± 0.07	88.34 ± 0.39	73.50 ± 0.32	88.31 ± 0.40
ROS	93.56 ± 0.05	77.03 ± 0.08	93.52 ± 0.05	88.15 ± 0.50	73.65 ± 0.40	88.10 ± 0.51
SMOTE	91.86 ± 0.08	77.50 ± 0.09	91.86 ± 0.08	87.06 ± 0.53	74.44 ± 0.39	86.92 ± 0.55
RUS	84.26 ± 0.14	64.93 ± 0.25	84.25 ± 0.14	83.69 ± 0.46	64.65 ± 0.38	83.67 ± 0.45

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.10 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **LR** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 87.53 (± 0.41), una BA de 87.54 (± 0.41) y un F1 de 70.40 (± 0.31).

Tabla 4.10
Validación cruzada de los modelos que utilizan TF-IGM y LR para Etiquetado A

Balanceo	Entrenamiento			Prueba		
	BA	F1 (M)	G-Mean	BA	F1 (M)	G-Mean
EDA	88.88 ± 0.05	71.18 ± 0.05	88.88 ± 0.05	87.54 ± 0.41	70.40 ± 0.31	87.53 ± 0.41
ROS	88.24 ± 0.05	70.05 ± 0.06	88.24 ± 0.05	86.89 ± 0.36	69.34 ± 0.33	86.88 ± 0.36
DEBOHID	88.14 ± 0.06	70.41 ± 0.06	88.13 ± 0.06	86.83 ± 0.41	69.73 ± 0.33	86.81 ± 0.41
SMOTE	86.31 ± 0.06	70.41 ± 0.07	86.29 ± 0.06	85.37 ± 0.41	69.82 ± 0.36	85.31 ± 0.42
RUS	76.46 ± 0.13	56.23 ± 0.18	76.41 ± 0.13	76.23 ± 0.57	56.09 ± 0.37	76.16 ± 0.56

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla **4.11** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **NB** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó DEBOHID como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 89.23 (± 0.35), una BA de 89.24 (± 0.35) y un F1 de 71.92 (± 0.30).

Tabla 4.11
Validación cruzada de los modelos que utilizan TF-IGM y NB para Etiquetado A

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
DEBOHID	91.20 ± 0.06	73.05 ± 0.06	91.16 ± 0.06	89.24 ± 0.35	71.92 ± 0.30	89.23 ± 0.35
EDA	92.28 ± 0.06	76.35 ± 0.06	92.27 ± 0.06	89.04 ± 0.47	74.37 ± 0.34	89.01 ± 0.48
ROS	91.34 ± 0.04	72.88 ± 0.06	91.30 ± 0.04	89.01 ± 0.36	71.60 ± 0.31	88.99 ± 0.36
SMOTE	90.79 ± 0.05	71.43 ± 0.08	90.72 ± 0.05	88.74 ± 0.34	70.34 ± 0.30	88.72 ± 0.33
RUS	89.96 ± 0.07	70.12 ± 0.21	89.89 ± 0.07	88.49 ± 0.35	69.45 ± 0.36	88.45 ± 0.35

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.12 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **SVM** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 88.74 (± 0.43), una BA de 88.77 (± 0.42) y un F1 de 74.00 (± 0.36).

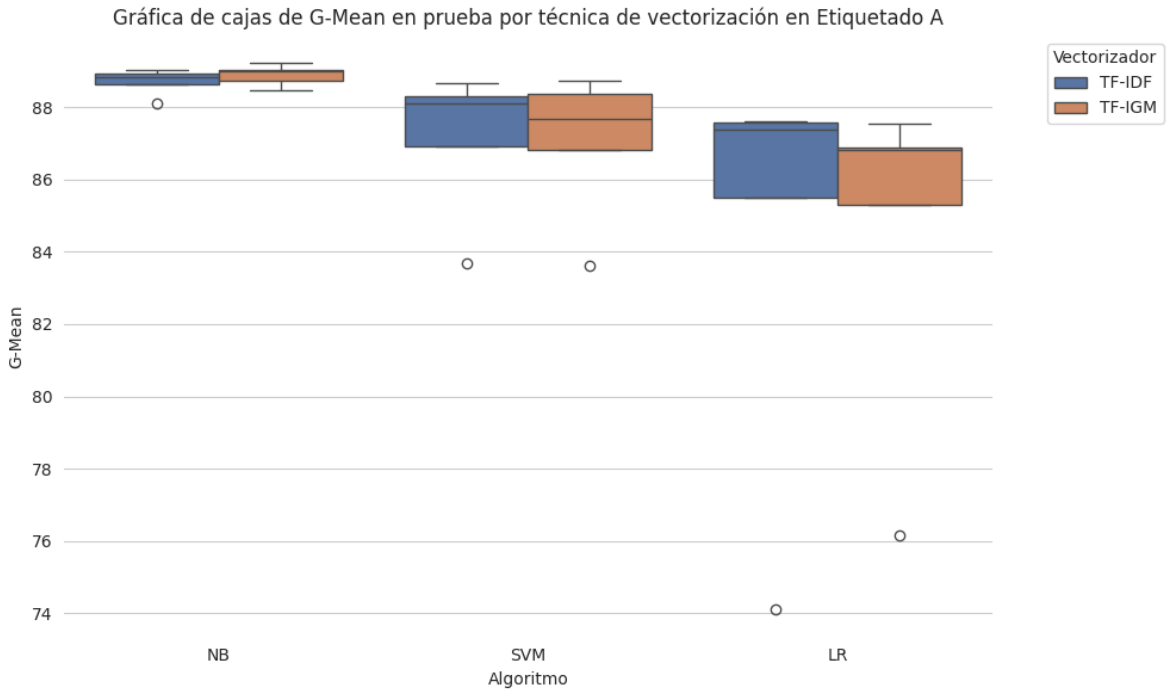
Tabla 4.12
Validación cruzada de los modelos que utilizan TF-IGM y SVM para Etiquetado A

Balanceo	Entrenamiento			Prueba		
	BA	F1 (M)	G-Mean	BA	F1 (M)	G-Mean
EDA	92.19 ± 0.07	76.18 ± 0.07	92.19 ± 0.07	88.77 ± 0.42	74.00 ± 0.36	88.74 ± 0.43
ROS	92.24 ± 0.06	75.05 ± 0.08	92.21 ± 0.06	88.39 ± 0.38	72.70 ± 0.33	88.37 ± 0.39
DEBOHID	91.08 ± 0.06	74.44 ± 0.06	91.08 ± 0.06	87.71 ± 0.41	72.35 ± 0.33	87.68 ± 0.42
SMOTE	90.35 ± 0.07	75.48 ± 0.08	90.35 ± 0.07	86.92 ± 0.42	73.34 ± 0.37	86.82 ± 0.44
RUS	84.22 ± 0.12	64.05 ± 0.20	84.19 ± 0.12	83.64 ± 0.47	63.77 ± 0.37	83.61 ± 0.46

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

Para realizar una comparación de los resultados obtenidos por cada método de extracción de características y por cada técnica de balanceo, a continuación se presenta el resultado del análisis de los mismos. En la Figura 4.15 se presenta una comparativa de los valores de G-Mean obtenidos en prueba. En esta visualización, los modelos están organizados por algoritmo de aprendizaje y agrupados según la técnica de extracción de características, sin distinguir entre técnicas de balanceo de clases.

Figura 4.15
Agrupación de resultados por técnica de vectorización en Etiquetado A



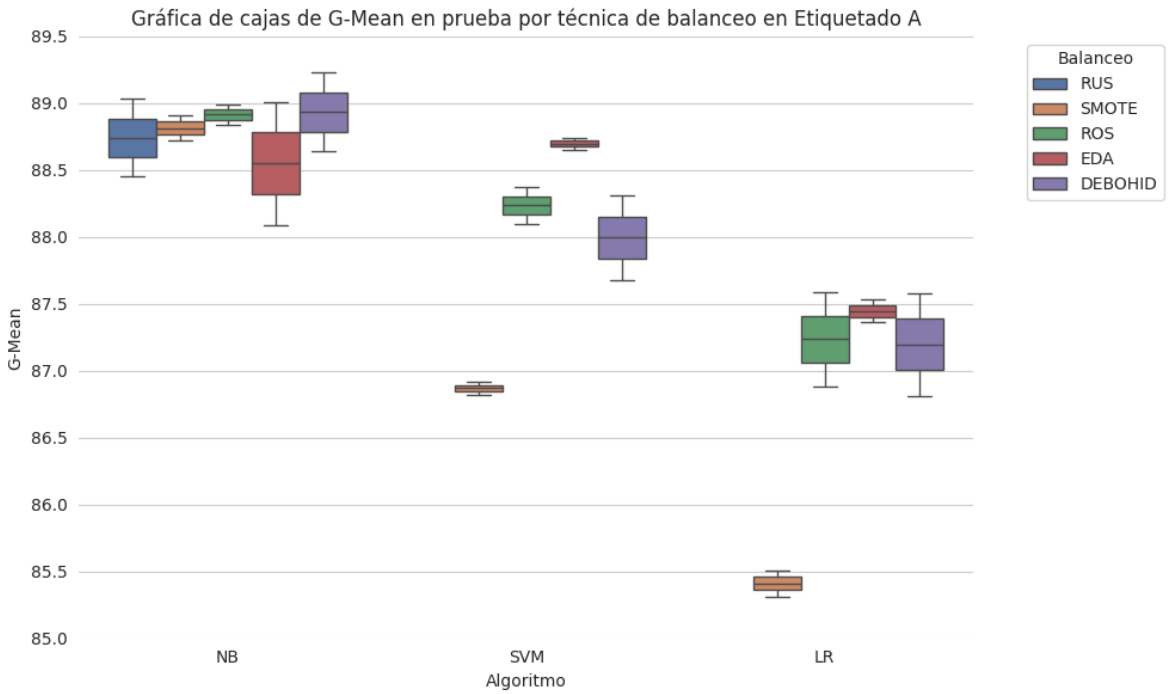
Nota. Se muestran los resultados en prueba de la validación cruzada realizada mediante SKCV, evaluados en términos de G-Mean y visualizados en un gráfico de cajas. Los datos están organizados por algoritmo y agrupados según la técnica de vectorización o extracción de características, sin distinguir entre las técnicas de balanceo de clases.

Se observa que el modelo con el mejor desempeño general es **NB** en combinación con **TF-IGM**, ya que presenta una mediana más alta, con todos sus valores por encima del 88 % y una baja variabilidad. En el caso de **SVM**, su mejor combinación se da con **TF-IDF**, debido a su mediana más alta y menor dispersión. De manera similar, **LR** también obtiene su mejor desempeño con **TF-IDF**, destacando por una mediana superior.

En contraste, para comparar las técnicas de balanceo en la Figura **4.16** se presentan los resultados de la validación cruzada realizada mediante SKCV para todos los modelos generados, utilizando los valores de G-Mean obtenidos en prueba y representados mediante una gráfica de cajas. En esta visualización, los modelos están organizados por algoritmo de aprendizaje y agrupados según la técnica de balanceo de clases, sin distinguir entre técnicas

de extracción de características.

Figura 4.16
Agrupación de resultados por técnica de balanceo de clases en Etiquetado A



Nota. Se muestran los resultados en prueba de la validación cruzada realizada mediante SKCV, evaluados en términos de G-Mean y visualizados en un gráfico de cajas. Los datos están organizados por algoritmo y agrupados según la técnica de balanceo de clases, sin distinguir entre las técnicas de extracción de características. Las cajas correspondientes a SVM/RUS y LR/RUS no se muestran, dado que sus valores fueron inferiores al 85 %.

Al igual que en la Figura 4.15, el modelo con el mejor desempeño general es **NB** en combinación con **DEBOHID**, ya que presenta los valores y la mediana más altos. Para **SVM**, la mejor combinación se da con **EDA**, puesto que alcanza los valores más altos y muestra la menor dispersión. Finalmente, **LR** también obtiene su mejor desempeño con **EDA**, pues esta técnica logró los resultados más altos y con menor variabilidad.

4.4.2 Evaluación del rendimiento de los modelos

A continuación, se evalúan los tres modelos de clasificación que obtuvieron los mejores resultados según la validación cruzada presentada en la sección anterior, de acuerdo a lo discutido en las Figuras **4.15** y **4.16**:

- **TF-IGM/DEBOHID/NB**. Modelo obtenido mediante Naïve Bayes, con extracción de características TF-IGM y DEBOHID como técnica de balanceo de datos.
- **TF-IDF/EDA/SVM**. Modelo basado en la Máquina de Soporte Vectorial, con extracción de características mediante TF-IDF y EDA como técnica de balanceo de datos.
- **TF-IDF/EDA/LR**. Modelo obtenido con Regresión Logística, utilizando TF-IDF para la extracción de características y EDA como técnica de balanceo de datos.

Estos modelos fueron entrenados con el conjunto de datos **HuatulcoResortReviews-A-Train** y evaluados utilizando el conjunto de datos **HuatulcoResortReviews-A-Test**. Seguidamente, se presentan los resultados de la evaluación de cada modelo, la cual se llevó a cabo mediante las métricas de **BA**, **Precisión**, **Sensibilidad**, **F1** y **G-Mean**.

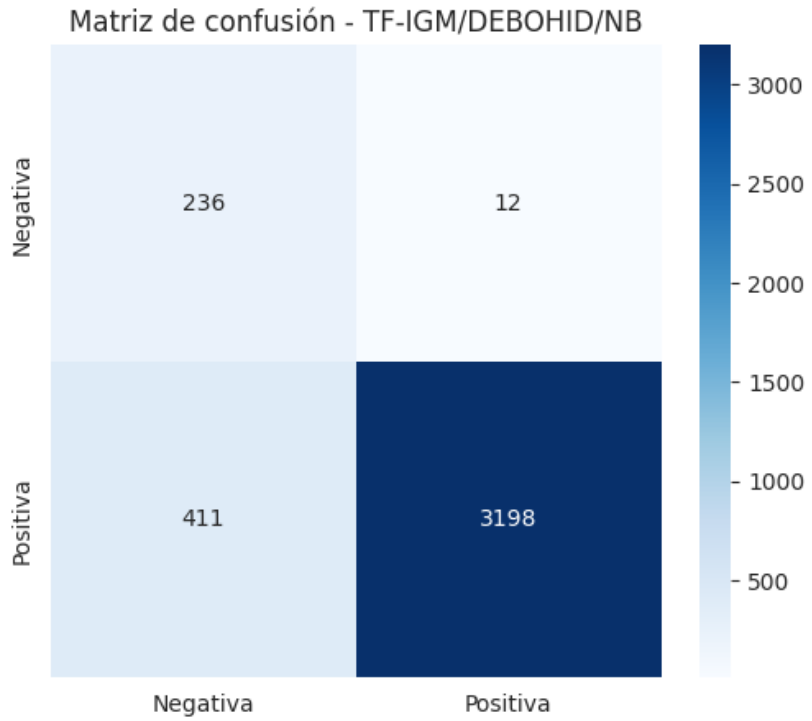
La Tabla 4.13 muestra los resultados de la evaluación del modelo que integra: **TF-IGM** como técnica de extracción de características; **DEBOHID** como técnica de balanceo de clases; y **NB** como algoritmo de aprendizaje. Por otro lado, la Figura 4.17 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 91.83, una BA de 91.89 y un F1 Macro de 73.27.

Tabla 4.13
Resultados de la evaluación del modelo TF-IGM/DEBOHID/NB para Etiquetado A

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		36.01	93.43	51.98	990
	Positiva		99.49	88.61	93.74	14437
	Prom. Macro		67.75	91.02	72.86	15427
	BA	91.02				15427
	G-Mean	90.99				15427
Prueba	Negativa		36.48	95.16	52.74	248
	Positiva		99.63	88.61	93.80	3609
	Prom. Macro		68.05	91.89	73.27	3857
	BA	91.89				3857
	G-Mean	91.83				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.17
Matriz de confusión del modelo TF-IGM/DEBOHID/NB para Etiquetado A



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

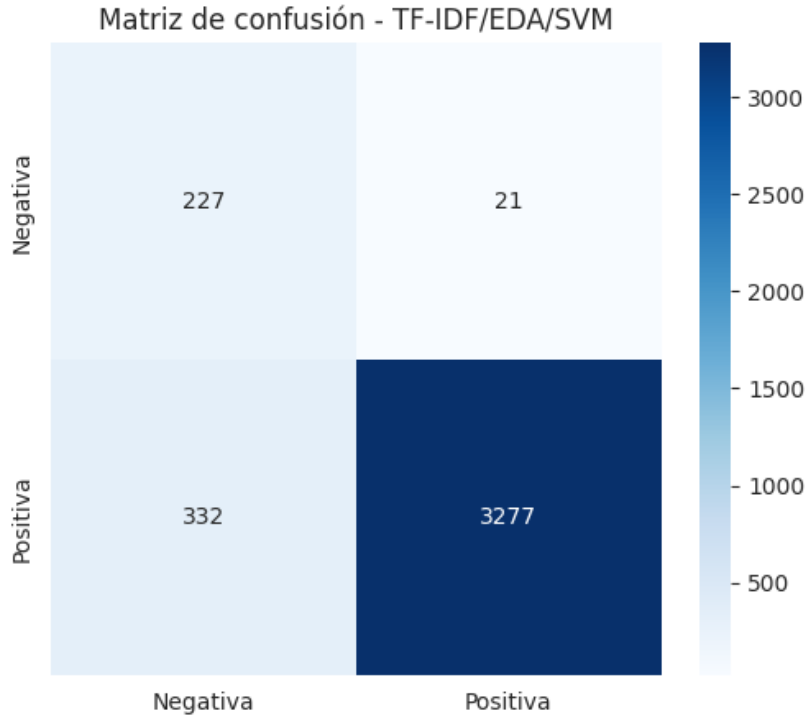
La Tabla 4.14 muestra los resultados de la evaluación del modelo que integra: **TF-IDF** como técnica de extracción de características; **EDA** como técnica de balanceo de clases; y **SVM** como algoritmo de aprendizaje. Por otro lado, la Figura 4.18 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 91.17, una BA de 91.17 y un F1 Macro de 75.57.

Tabla 4.14
Resultados de la evaluación del modelo TF-IDF/EDA/SVM para Etiquetado A

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		41.83	94.34	57.96	990
	Positiva		99.58	91.00	95.10	14437
	Prom. Macro		70.70	92.67	76.53	15427
	BA	92.67				15427
	G-Mean	92.66				15427
Prueba	Negativa		40.61	91.53	56.26	248
	Positiva		99.36	90.80	94.89	3609
	Prom. Macro		69.99	91.17	75.57	3857
	BA	91.17				3857
	G-Mean	91.17				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.18
Matriz de confusión del modelo TF-IDF/EDA/SVM para Etiquetado A



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

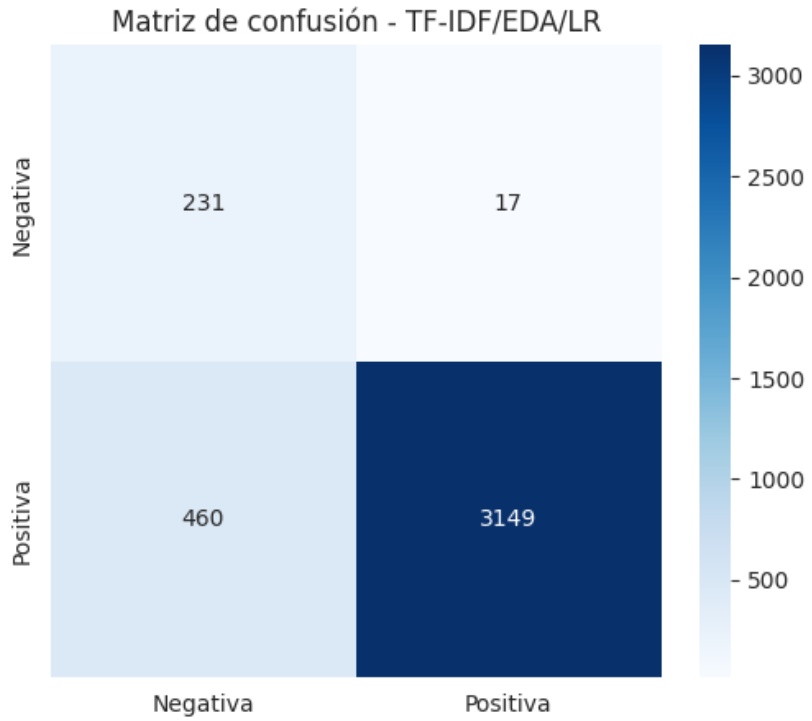
La Tabla 4.15 muestra los resultados de la evaluación del modelo que integra: **TF-IDF** como técnica de extracción de características; **EDA** como técnica de balanceo de clases; y **LR** como algoritmo de aprendizaje. Por otro lado, la Figura 4.19 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 90.20, una BA de 90.15 y un F1 Macro de 71.08.

Tabla 4.15
Resultados de la evaluación del modelo TF-IDF/EDA/LR para Etiquetado A

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		33.26	90.61	48.66	990
	Positiva		99.27	87.53	93.03	14437
	Prom. Macro		66.26	89.07	70.84	15427
	BA	89.07				15427
	G-Mean	89.06				15427
Prueba	Negativa		33.43	93.15	49.20	248
	Positiva		99.46	87.25	92.96	3609
	Prom. Macro		66.45	90.20	71.08	3857
	BA	90.15				3857
	G-Mean	90.20				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.19
Matriz de confusión del modelo TF-IDF/EDA/LR para Etiquetado A



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

4.4.3 Análisis y comparación de resultados

Para evaluar el rendimiento de los mejores modelos de clasificación y analizar el impacto de las distintas técnicas de preprocesamiento, se compararon sus resultados con aquellos de modelos a los que solo se les aplicó extracción de características, sin ningún otro tipo de procesamiento adicional. A estos últimos se les denominó modelos de referencia o *baseline*. En total, se generaron seis modelos *baseline*, combinando las dos técnicas de extracción de características: **TF-IDF** y **TF-IGM**, con los tres algoritmos de aprendizaje: **NB**, **SVM** y **LR**.

En la Tabla **4.16** se muestra la comparación entre los mejores modelos de cada algoritmo, que son producto de su combinación con las técnicas de extracción de características y balanceo de clases, contra los modelos *baseline*, en términos de BA, F1 Macro y G-Mean.

Tabla 4.16
Comparación de resultados de mejores modelos y modelos baseline para Etiquetado A

Modelo	BA	F1 Macro	G-Mean
TF-IGM/DEBOHID/NB	91.89	73.27	91.83
TF-IDF/EDA/SVM	91.17	75.57	91.17
TF-IDF/EDA/LR	90.15	71.08	90.20
TF-IDF/SVM	59.81	64.85	44.42
TF-IGM/SVM	54.41	56.56	29.78
TF-IDF/LR	52.02	52.28	20.08
TF-IGM/LR	50.67	49.82	12.68
TF-IDF/NB	50.00	48.34	00.00
TF-IGM/NB	50.00	48.34	00.00

Nota. Se muestran los resultados en prueba de los modelos expresados en Exactitud Balanceada (BA), F1 en Promedio Macro y G-Mean, se ordenan de mayor a menor según el desempeño respecto al G-Mean. Los modelos destacados en **negrita** corresponden a los mejores resultados obtenidos, analizados en la Sección **4.4.2**.

En primer lugar, los resultados muestran que los modelos que incorporan balanceo (**DEBOHID** y **EDA**) así como limpieza/selección de características superan ampliamente a los *baseline* en todas las métricas. El modelo **TF-IGM/DEBOHID/NB** demuestra el desempeño más sobresaliente en términos generales, alcanzando valores de BA y G-Mean cercanos al 92 % y un F1 Macro en torno al 73 %. No obstante, los otros dos modelos generados muestran un rendimiento comparable, con diferencias mínimas en relación con el modelo de mejor desempeño. Por otro lado, los modelos *baseline* presentan valores de G-Mean entre 0 % y 44 %, BA entre 50 % y 59 %, y un F1 Macro igualmente bajo, oscilando entre 48 % y 64 %.

Se observa, además, que los valores de BA y G-Mean son muy similares en los tres mejores modelos. Esto indica que las instancias, tanto positivas como negativas, se están recuperando de manera más equilibrada, lo que favorece la exactitud en la clasificación. Pese a ello, el F1 Macro es notablemente más bajo y no mostró un aumento considerable en comparación con estas métricas respecto a los modelos *baseline*, lo que sugiere que el

desbalance afecta principalmente la precisión y, en consecuencia, el F1-Score. Esto se refleja en las Tablas **4.13**, **4.14** y **4.15**, donde la precisión para la clase negativa oscila entre el 35 % y el 40 %.

Por lo tanto, el G-Mean es una métrica que ayuda a seleccionar el modelo más adecuado para identificar la polaridad de reseñas en inglés sobre nuestra base de datos de servicios de hoteles todo incluido.

4.5 Experimento 2: Clasificación de polaridad en HuatulcoResortReviews con Etiquetado B

Al igual que la Sección 4.4, tras especificar el preprocesamiento de la base de datos **HuatulcoResortReviews** ahora para el Etiquetado B, esta sección presenta la integración de la Sección 4.3 enfocada en el entrenamiento de modelos utilizando los algoritmos LR, NB y SVM.

Siguiendo los pasos descritos en la Etapa 3 del método propuesto en el capítulo 3 de esta tesis, la primera parte muestra los modelos obtenidos junto con su validación cruzada mediante SKCV. En total, se generaron treinta modelos que combinan las técnicas de vectorización o extracción de características TF-IDF y TF-IGM, junto con las técnicas de balanceo de clases: ROS, RUS, SMOTE, DEBOHID y EDA. Posteriormente, se lleva a cabo la evaluación y análisis de los resultados conforme a la Etapa 4 del método propuesto, empleando las métricas BA, G-Mean y F1-Score.

4.5.1 Entrenamiento de modelos de clasificación

En la etapa de entrenamiento, siguiendo el mismo enfoque que en la Sección 4.4.1, se empleó el conjunto de datos **HuatulcoResortReviews-B-Train**. El proceso de entrenamiento se llevó a cabo utilizando la biblioteca scikit-learn, de acuerdo con las especificaciones establecidas en la Sección 4.1. A continuación, se listan los algoritmos con sus respectivos parámetros:

- **LogisticRegression** (LR): $penalty = l1$, $C = 0.1$, $solver = saga$, $max\_iter = 1000$.
- **LinearSVC** (SVM): $penalty = l1$, $C = 0.03$, $dual = False$, $max\_iter = 1000$.
- **MultinomialNB** (NB): con los parámetros por defecto.

La Tabla **4.17** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **LR** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 69.22 (± 0.56), una BA de 69.76 (± 0.52) y un F1 de 60.97 (± 0.41).

Tabla 4.17
Validación cruzada de los modelos que utilizan TF-IDF y LR para Etiquetado B

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
EDA	73.79 ± 0.08	63.93 ± 0.07	73.56 ± 0.08	69.76 ± 0.52	60.97 ± 0.41	69.22 ± 0.56
DEBOHID	73.72 ± 0.09	64.27 ± 0.07	73.26 ± 0.09	69.82 ± 0.50	61.27 ± 0.41	69.02 ± 0.54
ROS	74.16 ± 0.08	64.28 ± 0.07	73.70 ± 0.08	69.66 ± 0.52	60.95 ± 0.42	68.82 ± 0.56
SMOTE	71.51 ± 0.09	63.32 ± 0.08	70.44 ± 0.10	68.15 ± 0.57	60.65 ± 0.47	66.71 ± 0.66
RUS	54.98 ± 0.33	48.80 ± 0.12	53.27 ± 0.35	54.70 ± 0.69	48.57 ± 0.49	52.80 ± 0.80

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.18 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **NB** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó RUS como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 69.51 (± 0.72), una BA de 69.72 (± 0.69) y un F1 de 59.00 (± 0.57).

Tabla 4.18
Validación cruzada de los modelos que utilizan TF-IDF y NB para Etiquetado B

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
RUS	76.64 ± 0.12	63.84 ± 0.18	76.56 ± 0.12	69.72 ± 0.69	59.00 ± 0.57	69.51 ± 0.72
DEBOHID	78.69 ± 0.10	67.05 ± 0.08	78.63 ± 0.10	69.47 ± 0.69	60.33 ± 0.57	69.06 ± 0.74
ROS	79.20 ± 0.08	67.09 ± 0.07	79.12 ± 0.08	69.45 ± 0.67	60.06 ± 0.52	69.04 ± 0.72
SMOTE	76.95 ± 0.07	64.54 ± 0.06	76.68 ± 0.07	68.97 ± 0.65	58.88 ± 0.54	68.55 ± 0.68
EDA	79.39 ± 0.08	69.31 ± 0.09	79.36 ± 0.09	68.34 ± 0.72	60.70 ± 0.61	67.58 ± 0.83

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla **4.19** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **SVM** entrenados, utilizando la extracción de características basada en **TF-IDF** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 68.47 (± 0.58), una BA de 69.50 (± 0.53) y un F1 de 61.38 (± 0.47).

Tabla 4.19
Validación cruzada de los modelos que utilizan TF-IDF y SVM para Etiquetado B

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
EDA	74.26 ± 0.09	65.00 ± 0.08	73.68 ± 0.10	69.50 ± 0.53	61.38 ± 0.47	68.47 ± 0.58
DEBOHID	73.92 ± 0.09	65.10 ± 0.08	72.91 ± 0.11	69.28 ± 0.50	61.48 ± 0.44	67.77 ± 0.55
ROS	74.62 ± 0.10	65.33 ± 0.08	73.64 ± 0.11	69.28 ± 0.58	61.28 ± 0.50	67.70 ± 0.64
SMOTE	72.12 ± 0.08	64.26 ± 0.07	70.63 ± 0.09	67.69 ± 0.52	60.78 ± 0.47	65.58 ± 0.63
RUS	56.71 ± 0.17	47.91 ± 0.26	50.27 ± 0.49	56.36 ± 0.46	47.70 ± 0.38	49.83 ± 0.61

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.20 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **LR** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 69.42 (± 0.57), una BA de 69.93 (± 0.54) y un F1 de 61.11 (± 0.44).

Tabla 4.20
Validación cruzada de los modelos que utilizan TF-IGM y LR para Etiquetado B

Balanceo	Entrenamiento			Prueba		
	BA	F1 (M)	G-Mean	BA	F1 (M)	G-Mean
EDA	73.52 ± 0.08	63.76 ± 0.05	73.29 ± 0.08	69.93 ± 0.54	61.11 ± 0.44	69.42 ± 0.57
DEBOHID	72.05 ± 0.10	62.58 ± 0.08	71.53 ± 0.10	69.01 ± 0.53	60.29 ± 0.42	68.17 ± 0.58
ROS	72.36 ± 0.11	62.23 ± 0.09	71.87 ± 0.11	68.89 ± 0.54	59.71 ± 0.43	68.09 ± 0.58
SMOTE	70.25 ± 0.09	61.69 ± 0.08	69.18 ± 0.10	67.21 ± 0.60	59.40 ± 0.49	65.81 ± 0.66
RUS	58.28 ± 0.13	47.74 ± 0.14	55.63 ± 0.16	57.94 ± 0.49	47.52 ± 0.37	55.14 ± 0.60

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla **4.21** muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **NB** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó ROS como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 69.01 (± 0.70), una BA de 69.30 (± 0.68) y un F1 de 59.57 (± 0.53).

Tabla 4.21
Validación cruzada de los modelos que utilizan TF-IGM y NB para Etiquetado B

Balanceo	**Entrenamiento**			**Prueba**		
	BA	**F1 (M)**	**G-Mean**	**BA**	**F1 (M)**	**G-Mean**
ROS	76.28 ± 0.08	64.57 ± 0.07	76.22 ± 0.08	69.30 ± 0.68	59.57 ± 0.53	69.01 ± 0.70
DEBOHID	75.85 ± 0.09	64.47 ± 0.08	75.79 ± 0.09	69.24 ± 0.64	59.72 ± 0.50	68.93 ± 0.67
RUS	73.29 ± 0.12	61.17 ± 0.17	73.28 ± 0.11	68.99 ± 0.68	58.18 ± 0.56	68.87 ± 0.70
EDA	78.97 ± 0.06	68.80 ± 0.06	78.92 ± 0.06	69.24 ± 0.68	61.27 ± 0.58	68.60 ± 0.76
SMOTE	74.75 ± 0.08	63.09 ± 0.07	74.53 ± 0.08	68.84 ± 0.67	58.93 ± 0.54	68.48 ± 0.69

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Tabla 4.22 muestra los resultados de la validación con SKCV, empleando $K = 10$ particiones para cada uno de los modelos **SVM** entrenados, utilizando la extracción de características basada en **TF-IGM** y cada una de las técnicas de balanceo: ROS, RUS, SMOTE, DEBOHID y EDA. SKCV se repitió $R = 5$ veces para calcular el CI al 95 %. El modelo con mejor desempeño fue el que utilizó EDA como técnica de balanceo de clases, alcanzando en prueba un G-Mean de 68.68 (± 0.53), una BA de 69.62 (± 0.49) y un F1 de 61.32 (± 0.42).

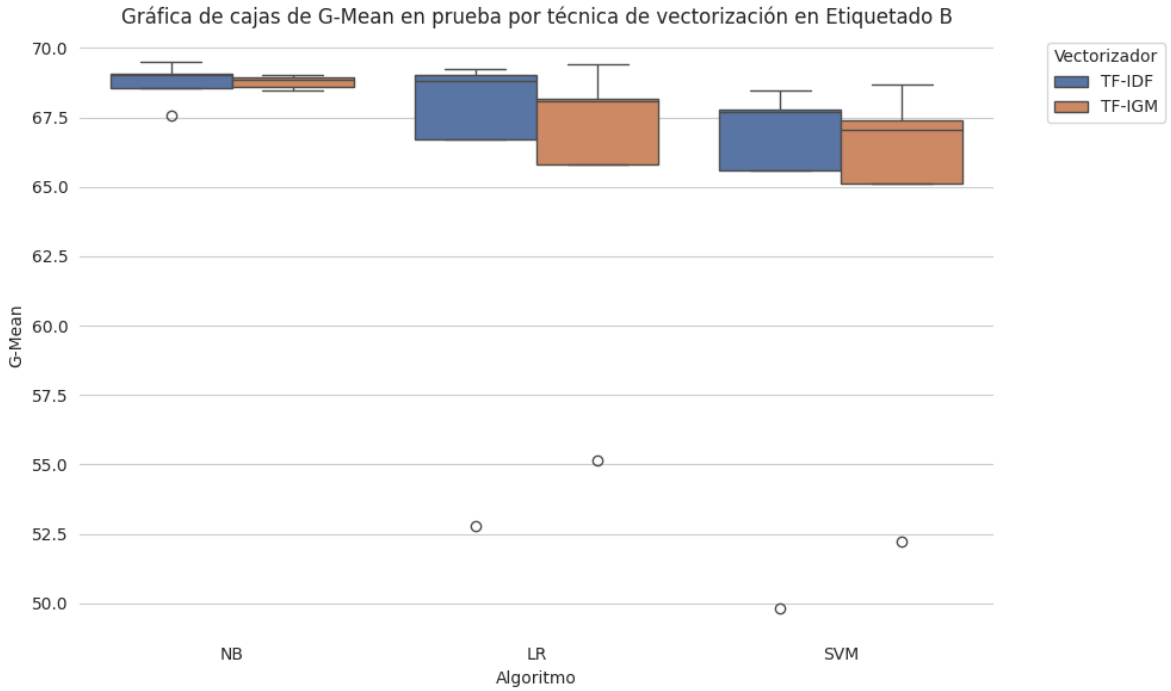
Tabla 4.22
Validación cruzada de los modelos que utilizan TF-IGM y SVM para Etiquetado B

Balanceo	Entrenamiento			Prueba		
	BA	F1 (M)	G-Mean	BA	F1 (M)	G-Mean
EDA	73.80 ± 0.08	64.46 ± 0.07	73.27 ± 0.09	69.62 ± 0.49	61.32 ± 0.42	68.68 ± 0.53
ROS	72.57 ± 0.09	63.01 ± 0.08	71.53 ± 0.10	68.89 ± 0.50	60.23 ± 0.42	67.39 ± 0.54
DEBOHID	72.06 ± 0.08	63.14 ± 0.07	70.90 ± 0.09	68.69 ± 0.53	60.52 ± 0.44	67.04 ± 0.60
SMOTE	70.45 ± 0.10	62.24 ± 0.09	68.91 ± 0.11	67.14 ± 0.55	59.69 ± 0.46	65.12 ± 0.64
RUS	58.61 ± 0.13	48.01 ± 0.16	53.05 ± 0.32	58.04 ± 0.47	47.60 ± 0.37	52.22 ± 0.62

Nota. Se muestran los resultados de la validación cruzada realizada mediante SKCV, empleando $K = 10$ particiones y $R = 5$ repeticiones. Para todos los modelos, se aplicó la misma limpieza de datos textuales y se seleccionó la misma cantidad de características. Los resultados, expresados en Exactitud Balanceada (BA), F1 Macro (M) y G-Mean, se ordenan de mayor a menor según el desempeño en prueba respecto al G-Mean. Se incluyen la media y el intervalo de confianza (CI) al 95 %.

La Figura **4.20** muestra los resultados de la validación cruzada realizada mediante SKCV para todos los modelos generados. Estos resultados se presentan a través de una gráfica de cajas, representando los valores de G-Mean obtenidos en prueba. En la visualización, los modelos se agrupan según la técnica de extracción de características y se organizan por algoritmo de aprendizaje, sin diferenciar entre las técnicas de balanceo de clases.

Figura 4.20
Agrupación de resultados por técnica de vectorización en Etiquetado B

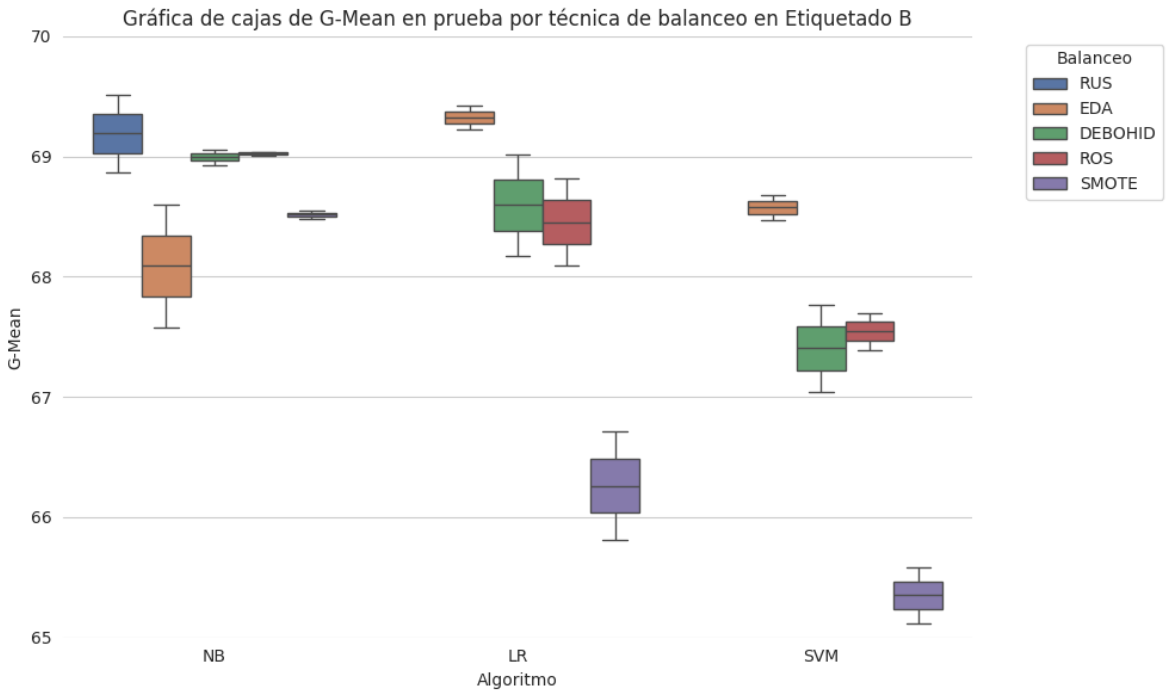


Nota. Se muestran los resultados en prueba de la validación cruzada realizada mediante SKCV, evaluados en términos de G-Mean y visualizados en un gráfico de cajas. Los datos están organizados por algoritmo y agrupados según la técnica de vectorización o extracción de características, sin distinguir entre las técnicas de balanceo de clases.

Se puede observar que el modelo con el mejor desempeño general es **NB** en combinación con **TF-IDF**, dado que presenta la mediana más alta y, en términos generales, resultados superiores. De igual manera, en el caso de **SVM**, su mejor combinación se da con **TF-IDF**, debido a su mediana más alta y menor dispersión. De manera análoga, **LR** alcanza su mejor desempeño cuando se utiliza **TF-IDF**.

Ahora, para comparar las técnicas de balanceo, en la Figura 4.21 se muestra la gráfica de cajas de los resultados de la validación cruzada realizada mediante SKCV para todos los modelos generados. Estos resultados representan los valores de G-Mean obtenidos en prueba. En la visualización, los modelos se agrupan según la técnica de balanceo de clases y se organizan por algoritmo de aprendizaje, sin diferenciar entre las técnicas de extracción de características.

Figura 4.21
Agrupación de resultados por técnica de balanceo de clases en Etiquetado B



Nota. Se muestran los resultados en prueba de la validación cruzada realizada mediante SKCV, evaluados en términos de G-Mean y visualizados en un gráfico de cajas. Los datos están organizados por algoritmo y agrupados según la técnica de balanceo de clases, sin distinguir entre las técnicas de extracción de características. Las cajas correspondientes a LR/RUS y SVM/RUS no se muestran, dado que sus valores fueron inferiores al 60 %.

Al igual que en la Figura 4.20, el modelo con el mejor desempeño general es **NB** en combinación con **RUS**, ya que presenta los valores y la mediana más altos. En el caso de **SVM**, la mejor combinación se obtiene con **EDA**, puesto que alcanza los valores más altos y muestra la menor dispersión. Finalmente, **LR** también obtiene su mejor desempeño con

EDA, pues esta técnica produjo los resultados más altos y con menor variabilidad.

4.5.2 Evaluación del rendimiento de los modelos

A continuación, se evalúan los tres modelos de clasificación que obtuvieron los mejores resultados según la validación cruzada presentada en la sección anterior, de acuerdo a lo discutido en las Figuras **4.20** y **4.21**:

- **TF-IDF/RUS/NB**. Modelo obtenido mediante Naïve Bayes, con extracción de características TF-IDF y RUS como técnica de balanceo de datos.
- **TF-IDF/EDA/SVM**. Modelo basado en la Máquina de Soporte Vectorial, con extracción de características mediante TF-IDF y EDA como técnica de balanceo de datos.
- **TF-IDF/EDA/LR**. Modelo obtenido con Regresión Logística, utilizando TF-IDF para la extracción de características y EDA como técnica de balanceo de datos.

Estos modelos fueron entrenados con el conjunto de datos **HuatulcoResortReviews-B-Train** y evaluados utilizando el conjunto de datos **HuatulcoResortReviews-B-Test**. Seguidamente, se presentan los resultados de la evaluación de cada modelo, la cual se llevó a cabo mediante las métricas de **BA**, **Precisión**, **Sensibilidad**, **F1** y **G-Mean**.

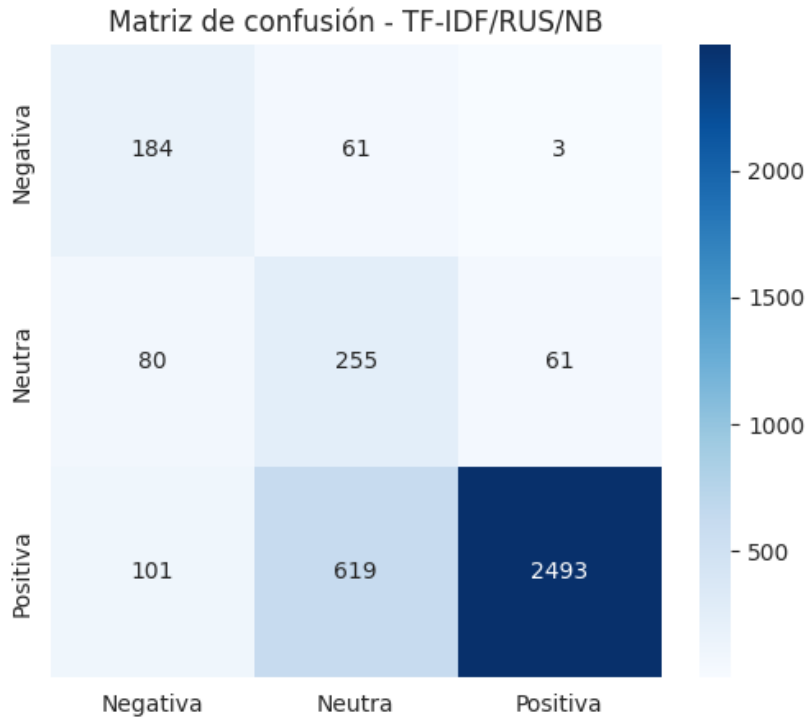
La Tabla 4.23 muestra los resultados de la evaluación del modelo que integra: **TF-IDF** como técnica de extracción de características; **RUS** como técnica de balanceo de clases; y **NB** como algoritmo de aprendizaje. Por otro lado, la Figura 4.22 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 71.84, una BA de 72.06 y un F1 Macro de 61.59.

Tabla 4.23
Resultados de la evaluación del modelo TF-IDF/RUS/NB para Etiquetado B

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		51.20	79.70	62.35	990
	Neutra		30.58	71.56	42.85	1586
	Positiva		98.41	77.92	86.97	12851
	Prom. Macro		60.06	76.39	64.06	15427
	BA	76.39				15427
	G-Mean	76.31				15427
Prueba	Negativa		50.41	74.19	60.03	248
	Neutra		27.27	64.39	38.32	396
	Positiva		97.50	77.59	86.41	3213
	Prom. Macro		58.39	72.06	61.59	3857
	BA	72.06				3857
	G-Mean	71.84				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.22
Matriz de confusión del modelo TF-IDF/RUS/NB para Etiquetado B



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

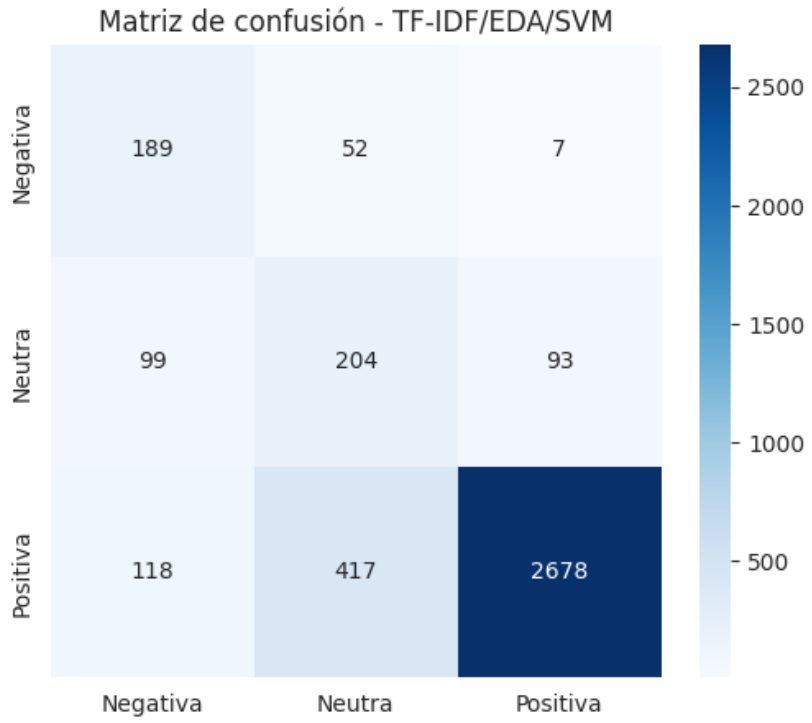
La Tabla 4.24 muestra los resultados de la evaluación del modelo que integra: **TF-IDF** como técnica de extracción de características; **EDA** como técnica de balanceo de clases; y **SVM** como algoritmo de aprendizaje. Por otro lado, la Figura 4.23 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 68.91, una BA de 70.36 y un F1 Macro de 61.79.

Tabla 4.24
Resultados de la evaluación del modelo TF-IDF/EDA/SVM para Etiquetado B

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		49.00	76.77	59.82	990
	Neutra		36.37	62.48	45.98	1586
	Positiva		97.34	84.46	90.44	12851
	Prom. Macro		60.90	74.57	65.41	15427
	BA	74.57				15427
	G-Mean	73.99				15427
Prueba	Negativa		46.55	76.21	57.80	248
	Neutra		30.31	51.52	38.17	396
	Positiva		96.40	83.35	89.40	3213
	Prom. Macro		57.75	70.36	61.79	3857
	BA	70.36				3857
	G-Mean	68.91				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.23
Matriz de confusión del modelo TF-IDF/EDA/SVM para Etiquetado B



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

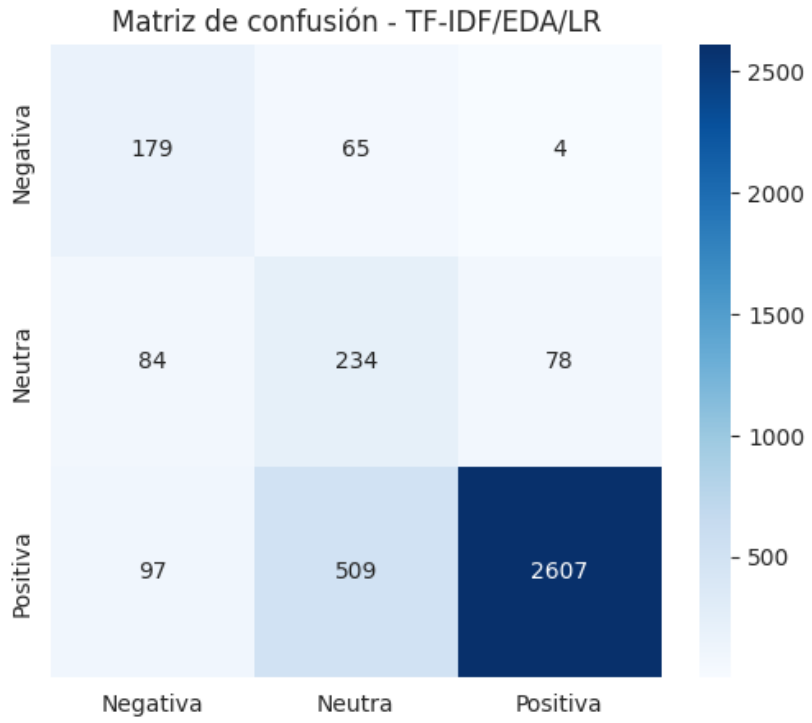
La Tabla 4.25 muestra los resultados de la evaluación del modelo que integra: **TF-IDF** como técnica de extracción de características; **EDA** como técnica de balanceo de clases; y **LR** como algoritmo de aprendizaje. Por otro lado, la Figura 4.24 muestra la matriz de confusión de este modelo. El modelo alcanzó en prueba un G-Mean de 70.21, una BA de 70.80 y un F1 Macro de 62.03.

Tabla 4.25
Resultados de la evaluación del modelo TF-IDF/EDA/LR para Etiquetado B

Conjunto			Precisión	Sensibilidad	F1	Ejemplos
Entrenamiento	Negativa		50.94	74.04	60.35	990
	Neutra		32.34	66.46	43.51	1586
	Positiva		97.63	81.51	88.85	12851
	Prom. Macro		60.30	74.00	64.24	15427
	BA	74.00				15427
	G-Mean	73.75				15427
Prueba	Negativa		49.72	72.18	58.88	248
	Neutra		28.96	59.09	38.87	396
	Positiva		96.95	81.14	88.34	3213
	Prom. Macro		58.54	70.80	62.03	3857
	BA	70.80				3857
	G-Mean	70.21				3857

Nota. Se muestran los resultados de la evaluación del modelo expresados en Precisión, Sensibilidad y F1 por clase, así como Exactitud Balanceada (BA), G-Mean y Promedio Macro. La tercera columna se reserva exclusivamente para los valores de BA y G-Mean, ya que son métricas globales.

Figura 4.24
Matriz de confusión del modelo TF-IDF/EDA/LR para Etiquetado B



Nota. Esta matriz de confusión corresponde a la evaluación con el conjunto de prueba.

4.5.3 Análisis y comparación de resultados

Para evaluar el rendimiento de los mejores modelos de clasificación y analizar el impacto de las distintas técnicas de preprocesamiento, se compararon sus resultados con los de modelos a los que únicamente se les aplicó extracción de características, sin ningún otro tipo de procesamiento adicional. A estos modelos, como se mencionó en la sección 4.4.3, se les denominó modelos de referencia o *baseline*. En total, se generaron seis modelos *baseline*, combinando las dos técnicas de extracción de características: **TF-IDF** y **TF-IGM**, con los tres algoritmos de aprendizaje: **NB**, **LR** y **SVM**.

En la Tabla 4.26 se muestra la comparación entre los mejores modelos de cada algoritmo, que son producto de su combinación con las técnicas de extracción de características

y balanceo de clases, contra los modelos *baseline*, en términos de BA, F1 Macro y G-Mean.

Tabla 4.26
Comparación de resultados de mejores modelos y modelos baseline para Etiquetado B

Modelo	BA	F1 Macro	G-Mean
TF-IDF/RUS/NB	72.06	61.59	71.84
TF-IDF/EDA/LR	70.80	62.03	70.21
TF-IDF/EDA/SVM	70.36	61.79	68.91
TF-IDF/LR	38.13	39.23	17.66
TF-IGM/LR	36.71	36.74	13.66
TF-IDF/SVM	37.11	37.36	11.95
TF-IGM/SVM	34.81	33.34	08.19
TF-IGM/NB	33.33	30.30	00.00
TF-IDF/NB	33.33	30.30	00.00

Nota. Se muestran los resultados en prueba de los modelos expresados en Exactitud Balanceada (BA), F1 en Promedio Macro y G-Mean, se ordenan de mayor a menor según el desempeño respecto al G-Mean. Los modelos destacados en **negrita** corresponden a los mejores resultados obtenidos, analizados en la Sección **4.5.2**.

En primer lugar, los resultados muestran que los modelos que incorporan balanceo (**RUS** y **EDA**) así como limpieza/selección de características superan significativamente a los *baseline* en todas las métricas evaluadas. En particular, el modelo **TF-IDF/RUS/NB** muestra el desempeño más destacado en términos generales, alcanzando valores de BA y G-Mean cercanos al 72 %, y un F1 Macro en torno al 61 %. No obstante, los otros dos modelos generados presentan un rendimiento comparable, con diferencias de apenas 1 % o 2 % respecto al modelo con mejor desempeño. En contraste, los modelos *baseline* presentan valores de G-Mean entre 0 % y 17 %, BA entre 33 % y 38 %, y un F1 Macro igualmente bajo, oscilando entre 30 % y 39 %.

Asimismo, se observa que los valores de BA y G-Mean son muy similares en los tres mejores modelos. Si bien esto sugiere que las instancias positivas, neutras y negativas se están recuperando de manera más equilibrada, los resultados no logran alcanzar valores más allá del 70 %. A su vez, al igual que en los resultados del Etiquetado A (ver Sección

4.4.3), el F1 Macro es inferior en comparación con las otras métricas, lo que indica que el elevado desbalance entre las clases afecta principalmente la precisión y, en consecuencia, el F1-Score. Esto se muestra en las Tablas **4.23**, **4.24** y **4.25**, donde las precisiones de las clases neutra y negativa varían entre el 30 % y el 50 %.

5 Conclusiones

Contenidos del Capítulo

5.1 Aportaciones 115
5.2 Trabajo a futuro 116

Este capítulo presenta las conclusiones derivadas del desarrollo de esta tesis. En ella se abordó el problema de seleccionar y evaluar técnicas que permitan maximizar el rendimiento de modelos de aprendizaje automático para clasificar la polaridad de reseñas en inglés sobre la experiencia de los clientes de servicios de hoteles todo incluido ubicados en Bahías de Huatulco, Oaxaca.

Se observó que la selección de la técnica de extracción de características, ya sea TF-IDF o TF-IGM, así como la técnica de balanceo de clases, depende en gran medida del enfoque de aprendizaje del algoritmo utilizado. El Clasificador Bayesiano Ingenuo o Naïve Bayes (NB), al tratarse de un modelo probabilístico, su desempeño general resultó similar con ambas técnicas de extracción de características; sin embargo, mostró un mejor rendimiento cuando se aplicaron técnicas de balanceo en el espacio vectorial, como DEBOHID.

En contraste, algoritmos paramétricos como Regresión Logística (LR) y Máquina de Soporte Vectorial (SVM), que se apoyan en funciones de optimización, se vieron más afectados por el desbalance en las clases. Estos algoritmos obtuvieron mejores resultados al emplear TF-IDF para la representación de características y fueron particularmente favorecidos por la técnica de balanceo basada en manipulación textual, EDA.

El procedimiento conducido, que corresponde al método propuesto, permitió alcanzar de manera satisfactoria el objetivo general establecido en el capítulo 1: “Clasificar la polaridad de reseñas en inglés sobre la experiencia de los clientes de servicios de hoteles todo incluido en Bahías de Huatulco, Oaxaca”. Para ello, se implementó una clasificación utilizando dos tipos de etiquetado: uno binario, que distinguió entre polaridades negativas y positivas; y otro multiclase, que incluyó las clases negativa, neutra y positiva.

Para alcanzar el objetivo general, fue necesario cumplir con cada uno de los objetivos específicos. El primero, relacionado con la revisión documental del estado del arte sobre técnicas de clasificación de polaridad en textos, se aborda en el capítulo 2, correspondiente al Marco Teórico. El segundo objetivo, que consistió en la construcción de un corpus de reseñas de hoteles todo incluido en inglés, se detalla en el capítulo 3, donde se explica el proceso de construcción del corpus, mientras que las reseñas recolectadas desde la plataforma Tripadvisor se presentan en el capítulo 4. En cuanto al tercer objetivo, en el capítulo 3 se propone el método que permitió explorar distintas técnicas de preprocesamiento y la generación de modelos de aprendizaje automático. Finalmente, los objetivos cuarto y quinto, que abarcan la evaluación de los modelos mediante métricas adecuadas y el análisis comparativo de los resultados, se desarrollan en el capítulo 4.

La investigación partió de la premisa de que, dada la diversidad de enfoques existentes, no hay una solución única que se adapte a todos los casos. Por ello, se realizó un análisis comparativo de diversas técnicas de extracción de características y balanceo de clases, utilizando distintas métricas, pero con especial énfasis en el G-Mean, para evaluar el desempeño de los modelos y garantizar un equilibrio adecuado en la identificación de la polaridad de las reseñas.

En relación con la hipótesis planteada en el capítulo 1: "El uso de una métrica como G-Mean, que equilibra la recuperación de las clases de polaridad positiva y negativa a comparación del F-Score, permitirá evaluar el impacto de las técnicas de extracción de características y balanceo de clases en modelos de aprendizaje automático para identificar la polaridad de reseñas en inglés sobre servicios de hoteles todo incluido."

La aplicación del método propuesto confirma que las técnicas de extracción de características y el balanceo de clases tienen un impacto significativo en la capacidad de los modelos para clasificar correctamente la polaridad de las reseñas. Como se expone en el capítulo 4, la métrica G-Mean resultó ser más adecuada que el F-Score o F1, ya que prioriza la exactitud general del modelo y penaliza fuertemente los casos en los que una o varias clases presentan una sensibilidad extremadamente baja o son clasificadas de forma incorrecta. En estos escenarios, el G-Mean refleja dicha deficiencia con valores cercanos a 0 %, como se muestra en las Tablas **4.16** y **4.26**.

Por otro lado, cuando todas las clases son reconocidas adecuadamente, el rendimiento del G-Mean mejora considerablemente. Esto se ve reflejado en los mejores modelos obtenidos: **TF-IGM/DEBOHID/NB** para el Etiquetado A y **TF-IDF/RUS/NB** para el Etiquetado B, que lograron resultados significativamente superiores a los modelos de referencia o *baseline*, alcanzando valores de G-Mean de 91.83 % y 71.84 %, respectivamente.

El F1, en cambio, al tratar de equilibrar precisión y sensibilidad, se ve más afectado por el desbalance de clases. Esto se observa en los modelos del Etiquetado A, donde la precisión para la clase minoritaria (negativa) no logra aumentar más allá de un 40 % (ver Sección **4.4.2**), y en el Etiquetado B, donde las clases minoritarias (negativa y neutra) no superan el 50 % de precisión (ver Sección **4.5.2**). Estos resultados pueden dar una impresión errónea sobre la capacidad de los modelos para clasificar correctamente todas las clases.

Asimismo, la Exactitud Balanceada (BA) y el G-Mean mostraron valores similares en los mejores modelos. Para el Etiquetado A, el modelo **TF-IGM/DEBOHID/NB** alcanzó valores de BA y G-mean cercanos al 92 %, mientras que para el Etiquetado B, el modelo **TF-IDF/RUS/NB** obtuvo valores alrededor del 72 %. En contraste, los modelos *baseline* tuvieron desempeños significativamente inferiores, con valores de BA de 50 % y G-Mean de 0 % en el Etiquetado A, y una BA de 33.33 % con un G-Mean de 0 % en el Etiquetado B. Estos resultados refuerzan la fuerte penalización que impone el G-Mean cuando los modelos no logran equilibrar la clasificación de todas las clases.

Por las razones expuestas en los párrafos anteriores, se puede concluir que los resultados obtenidos no proporcionan evidencia para rechazar la hipótesis planteada. Se confirma que el uso de G-Mean es más apropiado para evaluar la mejora en la clasificación de técnicas de extracción de características y balanceo de clases en modelos de aprendizaje automático entrenados con conjuntos de datos altamente desbalanceados.

5.1 Aportaciones

- Se creó un corpus de reseñas en inglés de hoteles todo incluido en Bahías de Huatulco, Oaxaca, obtenido de la plataforma Tripadvisor, que puede ser utilizado en futuras

investigaciones.

- Se propuso un método replicable para la clasificación de textos desbalanceados, integrando diferentes técnicas de preprocesamiento, modelos de aprendizaje automático y evaluación mediante distintas métricas.
- Se identificaron las combinaciones óptimas de técnicas de preprocesamiento que mejoran la exactitud de los modelos en la clasificación de polaridad de reseñas de hoteles todo incluido.
- Los resultados sugieren que la métrica G-Mean es más adecuada que F1 para evaluar la exactitud de los modelos en conjuntos de datos altamente desbalanceados, ya que penaliza el bajo desempeño en clases minoritarias.

5.2 Trabajo a futuro

Esta tesis se centró en el uso de enfoques tradicionales para la representación de texto y su clasificación, así como un etiquetado basado directamente en la puntuación otorgada por el usuario. Para investigaciones futuras, se sugiere la exploración de técnicas más avanzadas, entre las cuales se encuentran:

- Utilizar técnicas de representación de textos que tomen en cuenta representaciones semánticas y no únicamente frecuencias, por ejemplo *Word Embeddings* como *Word2Vec* o *GloVe*.
- Explorar modelos de clasificación más sofisticados basados en *Deep Learning*, como Redes Neuronales Recurrentes o *transformers* como *BERT*.
- Con el fin de reducir los errores de clasificación, investigar distintas técnicas de etiquetado para el conjunto de reseñas que no dependa de las puntuaciones, por ejemplo las basadas en enfoques de aprendizaje semi-supervisado y no supervisado.

Bibliografía

Barreda, A., & Bilgihan, A. (2013). An analysis of user-generated content for hotel experiences. *Journal of Hospitality and Tourism Technology*, *4*(3), 263-280. https://doi.org/10.1108/JHTT-01-2013-0001

Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*, 107134. https://doi.org/10.1016/j.knosys.2021.107134

Budhi, G. S., Chiong, R., Pranata, I., & Hu, Z. (2021). Using machine learning to predict the sentiment of online reviews: A new framework for comparative analysis. *Archives of Computational Methods in Engineering*, *28*(4), 2543-2566. https://doi.org/10.1007/s11831-020-09464-8

Burkov, A. (2023). *The hundred-page machine learning book* [OCLC: 1417057084]. Andriy Burkov.

Chang, V., Liu, L., Xu, Q., Li, T., & Hsu, C.-H. (2023). An improved model for sentiment analysis on luxury hotel review. *Expert Systems*, *40*(2), e12580. https://doi.org/10.1111/exsy.12580

Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Systems with Applications*, *66*, 245-260. https://doi.org/10.1016/j.eswa.2016.09.009

Cherdouh, S., Kherri, A., Abbaci, A., & Kebir, S. (2022). Using sentiment analysis of online hotel reviews to explore the effect of information and communication technologies on hotel guest satisfaction. *Journal of Tourismology*. https://doi.org/10.26650/jot.2022.8.1.1038566

Chowdhary, K. (2020). *Fundamentals of artificial intelligence*. Springer India. https://doi.org/10.1007/978-81-322-3972-7

Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer US. https://doi.org/10.1007/978-0-387-36797-2

Congreso del Estado Libre y Soberano de Oaxaca. (2021). *Derrama económica del sector turístico en el estado de Oaxaca* (Informe técnico). Centro de Estudios Económicos y Finanzas Públicas. Oaxaca, México. https://docs64.congresooaxaca.gob.mx/centros-estudios/CEEFP/estudiosCEEFP/6_DERRAMA_ECONOMICA_SECTOR_TURISTICO.pdf

Contreras Castañeda, E. D. (2021). La medición de la calidad del servicio en destinos turísticos: una revisión desde Colombia. *Innovar*, *31*(81), 35-48. https://doi.org/10.15446/innovar.v31n81.95571

Dharma, A. S., & Saragih, Y. G. R. (2022). Comparison of Feature Extraction Methods on Sentiment Analysis in Hotel Reviews. *Sinkron*, *7*(4), 2349-2354. https://doi.org/10.33395/sinkron.v7i4.11706

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Springer International Publishing. https://doi.org/10.1007/978-3-319-98074-4

Gazali Mahmud, F., Iman Hermanto, T., & Maruf Nugroho, I. (2023). Implementation Of K-Nearest Neighbor Algorithm With SMOTE For Hotel Reviews Sentiment Analysis. *SinkrOn*, *8*(2), 595-602. https://doi.org/10.33395/sinkron.v8i2.12214

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems* (Second edition). O'Reilly Media, Inc.

Hernández Martínez, I. (2022, 18 de noviembre). *Generación de prototipos usando un algoritmo genético* [Tesis de Maestría]. Universidad Autónoma del Estado de México. Consultado el 5 de septiembre de 2024, desde http://ri.uaemex.mx/handle/20.500.11799/138052

Kaya, E., Korkmaz, S., Sahman, M. A., & Cinar, A. C. (2021). DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets. *Expert Systems with Applications*, *169*, 114482. https://doi.org/10.1016/j.eswa.2020.114482

Khamphakdee, N., & Seresangtakul, P. (2021). Sentiment Analysis for Thai Language in Hotel Domain Using Machine Learning Algorithms. *Acta Informatica Pragensia*, *10*(2), 155-171. https://doi.org/10.18267/j.aip.155

Kharwal, A. (2020). *Bag of words in machine learning with python* [Thecleverprogrammer]. Consultado el 23 de octubre de 2024, desde https://thecleverprogrammer.com/2020/10/26/bag-of-words-in-machine-learning-with-python/

LeDoux, J. E. (2015). Feelings: What are they & how does the brain make them? *Daedalus*, *144*(1), 96-111. https://doi.org/10.1162/DAED_a_00319

Maldonado, C., & Hernández, G. (2011). *Guía para autogestión de CALIDAD servicios turísticos comunitarios* [OCLC: 1369471811]. OIT.

Márquez Reiter, R., Hidalgo Downing, R., & Iveson, M. (2023). Global Expectations, Local Realities: All-Inclusive Hotel Reviews and Responses on TripAdvisor. *Contrastive Pragmatics*, 1-33. https://doi.org/10.1163/26660393-bja10086

Mauri, A. G., & Minazzi, R. (2013). Web reviews influence on expectations and purchasing intentions of hotel potential customers. *International Journal of Hospitality Management*, *34*, 99-107. https://doi.org/10.1016/j.ijhm.2013.02.012

Meesad, P., Boonrawd, P., & Nuipian, V. (2011). A chi-square-test for word importance differentiation in text classification. *Proceedings of international conference on information and electronics engineering*, 110-114.

Mujahid, M., Kına, E., Rustam, F., Villar, M. G., Alvarado, E. S., De La Torre Diez, I., & Ashraf, I. (2024). Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering. *Journal of Big Data*, *11*(1), 87. https://doi.org/10.1186/s40537-024-00943-4

Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with Python: a guide for data scientists* (First edition). O'Reilly Media.

Polpinij, J., & Luaphol, B. (2021). Comparing of multi-class text classification methods for automatic ratings of consumer reviews [Series Title: Lecture Notes in Computer Science]. En P. Chomphuwiset, J. Kim & P. Pawara (Eds.), *Multi-disciplinary trends in artificial intelligence* (pp. 164-175, Vol. 12832). Springer International Publishing. https://doi.org/10.1007/978-3-030-80253-0_15

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 6086. https://doi.org/10.1038/s41598-024-56706-x

Sarkar, D. (2016). *Text analytics with python*. Apress. https://doi.org/10.1007/978-1-4842-2388-8

Satriaji, W., & Kusumaningrum, R. (2018). Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on Imbalanced Sentiment Analysis. *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, 1-5. https://doi.org/10.1109/ICICOS.2018.8621648

Secretaría de Turismo del Estado de Oaxaca. (2023). Boletín de Indicadores de la Actividad Turística 2023. https://www.oaxaca.gob.mx/sectur/wp-content/uploads/sites/65/2024/01/1.-Indicadores-de-la-Actividad-Turistica-2023.pdf

Shi, H.-X., & Li, X.-J. (2011). A sentiment analysis model for hotel reviews based on supervised learning. *2011 International Conference on Machine Learning and Cybernetics*, 950-954. https://doi.org/10.1109/ICMLC.2011.6016866

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). Natural Language Processing Advancements By Deep Learning: A Survey [Publisher: arXiv Version Number: 4]. https://doi.org/10.48550/ARXIV.2003.01200

Tosun, C., Dedeoğlu, B. B., & Fyall, A. (2015). Destination service quality, affective image and revisit intention: The moderating role of past experience. *Journal of Destination Marketing & Management*, *4*(4), 222-234. https://doi.org/10.1016/j.jdmm.2015.08.002

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731-5780. https://doi.org/10.1007/s10462-022-10144-1

Wei, J., & Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks [Version Number: 2]. https://doi.org/10.48550/ARXIV.1901.11196

Wibowo Haryanto, A., Kholid Mawardi, E., & Muljono. (2018). Influence of Word Normalization and Chi-Squared Feature Selection on Support Vector Machine (SVM) Text Classification. *2018 International Seminar on Application for Technology of Information and Communication*, 229-233. https://doi.org/10.1109/ISEMANTIC.2018.8549748

Yordanova, S., & Kabakchieva, D. (2017). Sentiment classification of hotel reviews in social media with decision tree learning. *International Journal of Computer Applications*, *158*(5), 1-7. https://doi.org/10.5120/ijca2017912806