Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas



TEMA:

Uso de aprendizaje automático para el análisis de sentimientos a través de los tweets de una empresa de alimentos: Caso de estudio Pizza Hut Ecuador.

AUTOR:

Ximena Lizeth Cáceres Castro

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN

QUITO, 2023 – 2024

DEDICATORIA

A mis amados padres, abuelos y familia, cuya presencia ha sido fundamental en cada uno de mis logros. Su infinito amor y paciencia han sido mi mayor fortaleza a lo largo de este camino. Agradezco de corazón su constante aliento y dedicación, que me ha impulsado a superarme día tras día.

A mi querido hermano, quien ha sido mi apoyo incondicional a lo largo de este trayecto. Su constante presencia, consejos y guía han sido fundamentales para llegar hasta aquí. Sin ti, este logro habría sido imposible.

A mis amigos de carrera, quienes han estado y siguen estando a mi lado, brindándome su apoyo. Su compañerismo y los mejores momentos compartidos durante este trayecto universitario han sido invaluables.

AGRADECIMIENTO

A toda mi familia, quienes han sido mi principal impulso para estudiar esta carrera. Su amor incondicional, apoyo constante y confianza en mí han sido el motor que me ha impulsado a alcanzar este logro. Gracias por estar a mi lado y creer en mí en cada paso del camino.

A mi novio Javier, por ser mi inspiración constante y por creer en mí cuando a veces yo misma dudaba. Su presencia en mi vida ha sido un regalo invaluable y le estaré eternamente agradecida por su amor incondicional.

A mi distinguido tutor, el Mgtr. Charles Escobar, quiero expresar mi más sincero agradecimiento por su valiosa enseñanza y los conocimientos impartidos. Su guía y dedicación han sido fundamentales para la culminación exitosa de este proyecto. Gracias por compartir su experiencia y por inspirarme a superar mis propios límites académicos.

RESUMEN

El presente proyecto de titulación se centra en el uso de técnicas de aprendizaje automático para realizar el análisis de sentimientos en los tweets de una empresa de alimentos, específicamente el caso de estudio de Pizza Hut Ecuador. El objetivo principal de este proyecto es emplear modelos de aprendizaje supervisado que sean capaces de clasificar los comentarios en categorías de sentimiento, tales como positivo, negativo o neutro.

Para alcanzar este objetivo, se comienza extrayendo los datos de la cuenta de Twitter de Pizza Hut Ecuador. A continuación, se lleva a cabo un proceso de limpieza y preparación de los datos, así como también se realiza el etiquetado necesario para entrenar los modelos.

Se implementan diversos modelos de aprendizaje supervisado, los cuales permiten clasificar los comentarios en las mencionadas categorías de sentimiento. Se evalúa el desempeño de estos modelos utilizando métricas como la precisión, la exactitud, la exhaustividad y el F1-score, con el fin de determinar cuál es el modelo que mejor se adapta a los datos.

ÍNDICE

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS VI

ÍNDICE DE FIGURAS VI

ÍNDICE DE TABLAS VII

CAPÍTULO I: INTRODUCCIÓN 1

1. MARCO DE REFERENCIA 1

1.1. Justificación 1

1.2. Planteamiento del problema 1

1.3. Objetivo General 2

1.4. Objetivos Específicos 2

1.5. Antecedentes 3

1.6. Alcance 4

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA 5

2. MARCO TEÓRICO 5

2.1. Pizza Hut 5

2.2. Twitter 5

2.3. Análisis de sentimientos 6

2.4. CRISP-DM 6

2.5. Herramientas y librerías 8

2.5.1. Web Scraping 8

2.5.1.1. Snscrape 8
2.5.2. Procesamiento de Lenguaje Natural (PLN) 8
2.5.3. Pandas 9
2.5.4. MatplotLib 9
2.5.5. RegEx 9
2.5.6. NLTK 10
2.5.7. Scikit-Learn 10
2.6. Aprendizaje automático 10
2.6.1. Aprendizaje supervisado y no supervisado 11
2.7. Modelos de aprendizaje supervisado 11
2.7.1. Bosques aleatorios (Random Forest) 11
2.7.2. Naive Bayes 12
2.7.3. Máquinas de vectores de soporte (SVM) 12
2.8. Métricas de evaluación 13
2.8.1. Matriz de confusión 13
2.8.2. Precisión 15
2.8.3. Exactitud 15
2.8.4. Recuerdo (Exahustividad) 15
2.8.5. F1-score (Puntuación F1) 15
CAPÍTULO III: METODOLOGÍA 17
3. Metodología de desarrollo 17

3.1. Investigación Cuantitativa....17
3.2. Metodología CRISP-DM....18
CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN....19
4. Desarrollo....19
4.1. Extracción de la data....19
4.2. Limpieza y preparación de los datos....20
4.2.1. Limpieza de los datos....20
4.3. Etiquetado de datos para el aprendizaje automático....22
4.4. Preparación de los datos....24
4.5. Modelado de datos con aprendizaje automático....27
4.5.1. Modelo Random Forest....28
4.5.2. Modelo SVM....29
4.5.3. Modelo Naive Bayes....30
4.6. Evaluación de los modelos....31
4.6.1. Modelo Random Forest Classifier....31
4.6.2. Modelo SVM....32
4.6.3. Modelo Naive Bayes....34
4.6.4. Comparación de los modelos....35
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES....37
5.1. Conclusiones....37
5.2. Recomendaciones....38

BIBLIOGRAFÍA 39

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

Figura 1 Matriz de Confusión 14

Figura 2 Código para la extracción de datos 19

Figura 3 Contenido de los tweets sin procesar 21

Figura 4 Función para la limpieza de los comentarios 22

Figura 5 *Modelo para predecir las etiquetas de los datos* 23

Figura 6 *Distribución de las clases* 26

Figura 7 Código Random Forest Classifier con datos de sobremuestreo 28

Figura 8 Código Random Forest Classifier con datos de submuestreo 29

Figura 9 Código SVC con datos de sobremuestreo 29

Figura 10 Código SVC con datos de submuestreo 30

Figura 11 Código Naive Bayes con datos de sobremuestreo 30

Figura 12 Código Naive Bayes con datos de submuestreo 31

Figura 13 Matriz de confusión Modelo Random Forest con datos sobremuestreados 31

Figura 14 Matriz de confusión Modelo Random Forest con datos submuestreados 32

Figura 15 Matriz de confusión Modelo SVM con datos sobremuestreados 33

Figura 16 Matriz de confusión Modelo SVM con datos submuestreados 33

Figura 17 Matriz de confusión Modelo Naive Bayes con datos sobremuestreados 34

Figura 18 Matriz de confusión Modelo Naive Bayes con datos submuestreados 35

ÍNDICE DE TABLAS

Tabla 1 Comparación de rendimiento de los modelos con datos sobremuestreados.....35

Tabla 2 Comparación de rendimiento de los modelos con los datos submuestreados.....36

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. Justificación

Twitter tiene gran acogida como una red social para poder expresar opiniones, agregar comentarios e incluso dar reseñas, por lo que es una de las herramientas más utilizadas para poder determinar y analizar las opiniones de los usuarios sobre distintos temas.

La creación de un modelo inteligente para este proyecto permitirá descubrir la aceptación que tiene Pizza Hut Ecuador por parte de los usuarios de Twitter que siguen su cuenta y comentan sus publicaciones. El modelo de análisis de sentimientos que se creará va a valorar la opinión como positiva, negativa o neutra.

Además, con este tipo de técnicas se puede beneficiar tanto a entidades públicas o privadas ya que de esta manera se puede conocer la aceptación por ejemplo de un producto o idea de una manera eficiente, aplicando técnicas de aprendizaje de máquina, ciencia de datos e inteligencia de negocios.

1.2. Planteamiento del problema

La generación de datos que se producen por medio de los usuarios de las redes sociales crece a diario a nivel mundial y es así como en los últimos años una de las redes sociales más usadas en Ecuador utilizadas para poder expresar pensamientos, emociones, opiniones es Twitter. Según las estadísticas hay una audiencia de 1.45 millones de usuarios, es decir, el 8% de la población de Ecuador y el 11% de los usuarios de Internet (Medina, 2022).

Dentro del país es limitado el uso de minería de datos para realizar predicciones en el comportamiento de los usuarios frente a temas económicos o políticos, comúnmente se utilizan

encuestas para medir el nivel de aceptación de una idea o producto. Pero si bien las encuestas permiten abarcar todos los aspectos de un tema, la falta de sinceridad o que los sentimientos y significados no se transmitan bien, pueden ser un problema al momento de analizar la aceptación de los encuestados a un producto, idea o servicio propuesto.

Dado esto, se puede definir la siguiente problemática principal:

- No existe una herramienta de aprendizaje automático con el fin de realizar el análisis de sentimientos en los comentarios de tweets de Pizza Hut.

Y los siguientes problemas secundarios:

- No se han etiquetado los datos generados de los comentarios de los tweets y tampoco han sido tratados por un modelo de aprendizaje.
- No se ha identificado un modelo los datos con aprendizaje automático para el análisis de sentimientos para el caso de estudio.
- No se ha evaluado los modelos que mejor se ajusten a los datos.

1.3. Objetivo General

Aplicar aprendizaje automático con el fin de realizar el análisis de sentimientos en los comentarios de tweets de Pizza Hut.

1.4. Objetivos Específicos

- Extraer la data desde la cuenta de Twitter de la empresa seleccionada.
- Preparar los datos para que puedan ser etiquetados y tratados por el modelo de aprendizaje.
- Etiquetar los datos con etiquetas de sentimiento (positivo, negativo o neutro), para el aprendizaje automático.
- Modelar los datos con aprendizaje automático para el análisis de sentimientos.

- Comparar diferentes modelos de aprendizaje para identificar el modelo que mejor se ajuste a los datos.

1.5. Antecedentes

La aplicación del análisis de sentimientos en Twitter para evaluar la opinión de los usuarios sobre una empresa o marca se ha convertido en una técnica cada vez más popular en la investigación de mercados y en la toma de decisiones empresariales. En particular, el uso del aprendizaje automático para el análisis de sentimientos en Twitter ha demostrado ser una herramienta eficaz para extraer información valiosa de grandes volúmenes de datos.

En particular, en el sector de la alimentación, el análisis de sentimientos en las redes sociales puede ser de gran utilidad para evaluar la satisfacción del cliente y mejorar la calidad de los productos. Sin embargo, hasta el momento, hay pocos estudios que hayan aplicado técnicas de aprendizaje automático para el análisis de sentimientos en empresas de alimentos.

Pero existen algunas investigaciones aplicando análisis de sentimientos en otras áreas y empresas. Por ejemplo, en un estudio realizado por (Saura et al., 2018), se utilizó el análisis de sentimientos en Twitter para medir el sentimiento del cliente en relación al evento Black Friday y de esta forma evaluar la satisfacción del cliente ante las promociones de ofertas en redes sociales. Los resultados mostraron que el análisis de sentimientos en Twitter puede ser una herramienta eficaz para mejorar la calidad de los productos y servicios.

En otro estudio realizado por, (López Fierro & Pacheco Villamar, 2021), se aplica análisis de sentimientos para medir la aceptación de la ciudadanía y conocer la opinión general, durante el debate y vueltas presidenciales ecuatorianas del 2021. Con los resultados se evidenció que estas técnicas pueden beneficiar tanto a entidades públicas como privadas para conocer la aprobación de una idea, producto o servicio.

En este contexto, la aplicación de técnicas de aprendizaje automático para el análisis de sentimientos en los comentarios de tweets de Pizza Hut Ecuador es un área de investigación relevante y novedosa. Al analizar la opinión de los usuarios de Twitter sobre esta empresa de alimentos, se puede obtener información valiosa sobre la satisfacción del cliente, las fortalezas y debilidades de la marca y las oportunidades de mejora.

1.6. Alcance

Este proyecto de titulación tiene como alcance el uso de un modelo inteligente para la clasificación de sentimientos en Twitter sobre los comentarios de los tweets de Pizza Hut Ecuador, que tomará en cuenta los sentimientos positivos, negativos o neutros, mismos que serán etiquetados en cada comentario cumpliendo las características que se necesiten y cuyo objetivo es entrenar algoritmos de aprendizaje de máquina y de esta manera escoger el que mejor se ajuste a los datos.

Para poder recopilar los datos se hará uso de scraping, usando como leguaje de programación Python, de esta manera se obtendrá datos en un formato JSON que tendrán su tratamiento para poder usarlos y trabajar con estos.

Después de la recopilación de datos y de ser procesados por el modelo es necesario poder evidenciar si el modelo funciona de forma adecuada, por lo que realizarán varias pruebas al modelo entrenado.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. MARCO TEÓRICO

2.1. Pizza Hut

Pizza Hut es una reconocida cadena de restaurantes de comida rápida especializada en pizzas, que tiene presencia a nivel internacional. La historia de la empresa se remonta al año 1958 cuando dos hermanos abrieron su primer establecimiento en Wichita (Kansas), se expandieron y lograron que la marca destacara, siendo así que en el 2006 en Ecuador se innovó la forma de ver y consumir pizza con Hut Cheese. Años más tarde con el principio del Call Center fueron pioneros en el servicio a domicilio.

Desde entonces, Pizza Hut se ha convertido en una opción popular para los amantes de la pizza en Ecuador, ofreciendo una amplia variedad de sabores y opciones de personalización. A lo largo de los años, Pizza Hut Ecuador ha seguido expandiendo su presencia en el país, abriendo nuevos restaurantes en diferentes ciudades y localidades.

2.2. Twitter

Las redes sociales son plataformas en línea que permiten a los usuarios compartir información, interactuar y conectarse con otros usuarios. Según Boyd y Ellison (2007), "Las redes sociales han transformado la forma en que la gente se comunica y comparte información". En particular, Twitter se ha convertido en una plataforma muy popular para compartir opiniones, noticias y comentarios sobre diversos temas.

El análisis de sentimientos en Twitter es una técnica que se enfoca en el procesamiento de los datos de los usuarios, principalmente en los textos de los tweets, para extraer información sobre las emociones, actitudes y opiniones de los usuarios acerca de una marca, producto o servicio. Esta técnica se basa en el procesamiento de lenguaje natural, en donde se utilizan

algoritmos de aprendizaje automático para clasificar los tweets según su polaridad (positiva, negativa o neutral).

2.3. Análisis de sentimientos

El análisis de sentimientos, también conocido como minería de opinión, es una técnica de procesamiento de lenguaje natural que se utiliza para identificar y categorizar las emociones expresadas en un texto. Según Pang y Lee (2008), consiste en "determinar la polaridad de un texto, es decir, si el texto expresa una opinión positiva, negativa o neutral".

Esta técnica se ha aplicado en diversas áreas, como la investigación de mercados, la gestión de la reputación en línea y el análisis de comentarios en redes sociales. En el caso de las empresas, el análisis de sentimientos se utiliza para evaluar la percepción que tienen los clientes sobre la marca, productos o servicios, con el fin de tomar decisiones estratégicas en cuanto a marketing, publicidad, atención al cliente, entre otros.

Existen diversas herramientas y algoritmos que permiten realizar análisis de sentimientos de manera automatizada. Según Liu (2012), estas herramientas se basan en diferentes enfoques, como el análisis léxico, el aprendizaje automático y la minería de datos. Sin embargo, es importante tener en cuenta que estas herramientas no siempre son precisas y pueden generar errores en la clasificación de las emociones expresadas en un texto. Por lo tanto, se recomienda combinar el análisis automatizado con la revisión manual para garantizar resultados óptimos del modelo desarrollado.

2.4. CRISP-DM

Cross-Industry Standard Process for Data Mining o por sus siglas CRISP-DM es un modelo de proceso estándar utilizado en el campo de la minería de datos. Fue desarrollado por un consorcio de expertos en la industria y se ha convertido en un marco de referencia ampliamente aceptado para el desarrollo de proyectos de minería de datos.

El modelo CRISP-DM consta de seis etapas principales, que son:

- **Comprensión del negocio:** En esta etapa inicial, se busca comprender los objetivos del negocio y los requisitos del proyecto. Se definen los objetivos de la minería de datos y se establece un plan general para el proyecto.
- **Comprensión de los datos:** En esta fase, se realiza una exploración de los datos disponibles para comprender su estructura, calidad, distribución y relaciones. Se identifican las fuentes de datos relevantes y se realizan tareas de preprocesamiento para preparar los datos para su análisis.
- **Preparación de los datos:** Para esta etapa, se seleccionan, limpian y transforman los datos para que sean adecuados para el análisis. Se aplican técnicas de limpieza, integración y selección de variables para garantizar la calidad de los datos.
- **Modelado:** En esta fase, se construyen modelos de minería de datos utilizando técnicas específicas, como árboles de decisión, regresión, redes neuronales, entre otros. Además se ajustan y evalúan los modelos para encontrar el mejor rendimiento.
- **Evaluación:** En esta etapa, se evalúa el rendimiento de los modelos construidos utilizando métricas y técnicas de validación. Se verifica si los modelos cumplen con los objetivos del proyecto y se determina su capacidad de generalización y aplicabilidad.
- **Despliegue:** En esta fase final, se implementan los modelos seleccionados en un entorno de producción y se realiza el seguimiento de su desempeño en tiempo real. Se generan informes y se comunican los resultados a las partes interesadas.

De acuerdo con IBM (2021), este modelo proporciona una estructura sistemática, flexible y repetible para el desarrollo de proyectos de minería de datos. Permite una gestión efectiva de los proyectos y promueve la colaboración entre los equipos de negocio y los expertos en minería de datos.

2.5. Herramientas y librerías

2.5.1. Web Scraping

Desde el punto de vista de Espin Riofrio et al., (2021) “es el proceso de recolectar datos contenidos en páginas web mediante técnicas automatizadas”. Esta técnica se ha vuelto cada vez más popular debido al gran volumen de información que se encuentra en línea y a la necesidad de procesar y analizar esta información de manera efectiva.

Algunas de las herramientas más utilizadas para el web scraping son Scrapy, BeautifulSoup, Selenium y Snscrape. Para este proyecto se va a utilizar Snscrape para la extracción de los datos de la cuenta de Twitter de PizzaHutEC.

2.5.1.1. Snscrape

Snscrape es una herramienta basada en Python que permite la extracción de datos de Twitter sin la necesidad de una API. Esta herramienta permite la búsqueda de tweets a través de palabras clave, ubicación, usuarios y otros filtros, lo que permite una extracción específica de los datos de interés. Además, proporciona opciones flexibles para el almacenamiento de datos extraídos, lo que facilita su posterior análisis.

2.5.2. Procesamiento de Lenguaje Natural (PLN)

El análisis de sentimientos se puede realizar a través de diferentes técnicas y herramientas, incluyendo el procesamiento de lenguaje natural, el aprendizaje automático y la minería de datos. Según Jurafsky y Martin (2019), "El procesamiento de lenguaje natural es una de las aplicaciones más importantes de la inteligencia artificial y se utiliza en una amplia variedad de tareas, como la traducción automática, el reconocimiento de voz y la clasificación de texto".

Tiene una amplia gama de aplicaciones en diferentes industrias, como la salud, finanzas y educación, entre otras, lo que lo convierte en una herramienta poderosa para el análisis de datos y la toma de decisiones.

En este sentido, se puede mencionar como una herramienta importante para la comprensión de grandes cantidades de información y además, como un enfoque clave en el análisis de sentimientos, ya que permite analizar el lenguaje humano en términos de estructura, significado y contexto.

2.5.3. Pandas

De acuerdo con Mckinney (2018), Pandas es una librería de código abierto en Python diseñada con el objetivo de facilitar la manipulación y análisis de datos de alto rendimiento. Proporciona numerosas funciones y métodos que permiten llevar a cabo operaciones de transformación de datos, como filtrado, selección, agregación y limpieza. Pandas se ha convertido en una herramienta ampliamente utilizada debido a su capacidad para simplificar el trabajo con datos en Python.

2.5.4. MatplotLib

Según Hunter (2007), la librería de visualización de datos en Python es una herramienta ampliamente utilizada que ofrece una amplia gama de herramientas y funciones para crear gráficos estáticos, interactivos y de alta calidad. Esta librería es reconocida por su flexibilidad y capacidad para crear visualizaciones personalizadas, lo que la convierte en una de las más populares en su campo.

2.5.5. RegEx

Es un módulo que proporciona funcionalidades para trabajar con expresiones regulares. Según Goyvaerts y Levithan (2012), una expresión regular es una secuencia de caracteres que define un patrón de búsqueda. Esta herramienta poderosa y versátil se utiliza para realizar búsquedas, procesamiento y manipulación de texto.

2.5.6. NLTK

NLTK (Natural Language Toolkit) es una librería en Python que proporciona una amplia variedad de herramientas y recursos para el procesamiento del lenguaje natural (PLN). De acuerdo con Bird et al. (2009), es reconocida como una de las bibliotecas más populares y utilizadas debido a su facilidad de uso y su extensa colección de módulos y datos lingüísticos. Entre sus características principales se encuentran la tokenización, el etiquetado de partes del discurso, el análisis sintáctico, el etiquetado, el stemming y la lematización.

2.5.7. Scikit-Learn

Pedregosa et al. (2011) describen scikit-learn, también conocido como sklearn, como una librería en Python especializada en aprendizaje automático. Esta librería ofrece una amplia variedad de algoritmos, herramientas y funciones para realizar implementaciones eficientes de algoritmos de aprendizaje supervisado y no supervisado.

Entre los algoritmos que se pueden abordar con sklearn se encuentran la clasificación, regresión, clustering, reducción de dimensionalidad y selección de características. Debido a su versatilidad y capacidad para trabajar con datos de manera eficiente, sklearn se ha convertido en una opción ampliamente utilizada para realizar tareas de aprendizaje automático en Python.

2.6. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial que permite a los sistemas informáticos aprender y mejorar por sí solos a partir de datos. En el contexto del análisis de sentimientos, el aprendizaje automático permite que los sistemas puedan identificar patrones y relaciones en los datos, y así poder clasificar los comentarios en positivos, negativos o neutros de manera precisa.

Según un estudio de Cámara, E. M., et al. (2011), el aprendizaje automático ha mostrado resultados prometedores en el análisis de sentimientos, ya que ha mostrado ser una herramienta

eficaz para clasificar comentarios en positivos, negativos o neutros. El aprendizaje automático se divide en dos categorías: supervisado y no supervisado.

2.6.1. Aprendizaje supervisado y no supervisado

En el aprendizaje supervisado, el sistema se entrena con un conjunto de datos previamente etiquetados para aprender a reconocer patrones y relaciones entre los datos. Por otro lado, en el aprendizaje no supervisado, el sistema busca patrones en los datos sin la necesidad de que los datos estén etiquetados.

En el contexto del análisis de sentimientos, el aprendizaje supervisado es el enfoque más utilizado, ya que permite que el sistema aprenda de los datos previamente etiquetados y, por lo tanto, pueda clasificar nuevos comentarios de manera precisa.

2.7. Modelos de aprendizaje supervisado

Los modelos de aprendizaje automático supervisado son aquellos que se entrenan utilizando datos etiquetados, es decir, datos que tienen una respuesta o variable objetivo conocida. A continuación, se presentan los modelos que se van a utilizar para el caso de estudio planteado:

2.7.1. Bosques aleatorios (Random Forest)

Según la perspectiva de Martinez Heras (2020), es un modelo que se basa en la idea de combinar varios árboles de decisión para mejorar la precisión de las predicciones. Cada árbol de decisión se entrena con muestras aleatorias de los datos de entrenamiento, y luego las predicciones se combinan para obtener un resultado final.

En el caso específico de Pizza Hut Ecuador, el uso de Random Forest permitiría una clasificación precisa de los comentarios de los usuarios en términos de sentimientos positivos, negativos o neutros. Al entrenar el modelo con un conjunto de datos etiquetados previamente, el modelo puede aprender patrones y características relevantes que contribuyen a la percepción de los usuarios sobre la empresa.

2.7.2. Naive Bayes

Este modelo se basa en el teorema de Bayes y asume independencia condicional entre las características. A pesar de su simplicidad, ha demostrado ser efectivo en una variedad de aplicaciones, incluyendo el procesamiento del lenguaje natural y la minería de textos.

El funcionamiento este modelo se fundamenta en el cálculo de probabilidades condicionales utilizando el teorema de Bayes. Dado un conjunto de características, el modelo calcula la probabilidad de que un texto o instancia pertenezca a una determinada clase o categoría. Utilizando estas probabilidades condicionales, se puede determinar la clase más probable para una nueva instancia.

Según Jurafsky y Martin (2019), una de las ventajas del modelo Naive Bayes es su capacidad para manejar eficientemente datos de alta dimensionalidad y características categóricas. Además, puede ser fácilmente entrenado con conjuntos de datos etiquetados y su implementación es relativamente sencilla.

2.7.3. Máquinas de vectores de soporte (SVM)

Según Martinez Heras (2020), es un algoritmo de aprendizaje supervisado utilizado para la clasificación y regresión. Se basa en la idea de encontrar un hiperplano óptimo que separe las muestras de diferentes clases en un espacio de características. El objetivo del SVM es encontrar el hiperplano que maximiza el margen entre las clases, lo que significa que busca la mejor separación posible entre los datos.

En términos sencillos, SVM busca construir una línea o un plano que separe los datos en diferentes categorías. Este algoritmo puede ser útil en problemas de clasificación binaria y también se puede extender para problemas de clasificación multiclase. SVM es especialmente eficaz en conjuntos de datos con características no lineales, ya que utiliza funciones de kernel

para mapear los datos a un espacio dimensional superior donde puedan ser separados de manera más efectiva.

2.8. Métricas de evaluación

Las métricas de evaluación en el contexto del aprendizaje automático son medidas utilizadas para evaluar el rendimiento y la efectividad de un modelo. Estas métricas permiten cuantificar y comparar el desempeño del modelo en función de su capacidad para realizar predicciones precisas y útiles sobre los datos de prueba.

Además, se puede emplear diversas métricas para evaluar distintos aspectos del rendimiento de un modelo. en el caso de este proyecto, se utilizarán la matriz de confusión, la precisión, la exhaustividad, la exactitud, el F1-score. Estas métricas brindan una visión objetiva de cómo se está desempeñando el modelo y pueden ayudar a tomar decisiones informadas sobre su utilidad y eficacia.

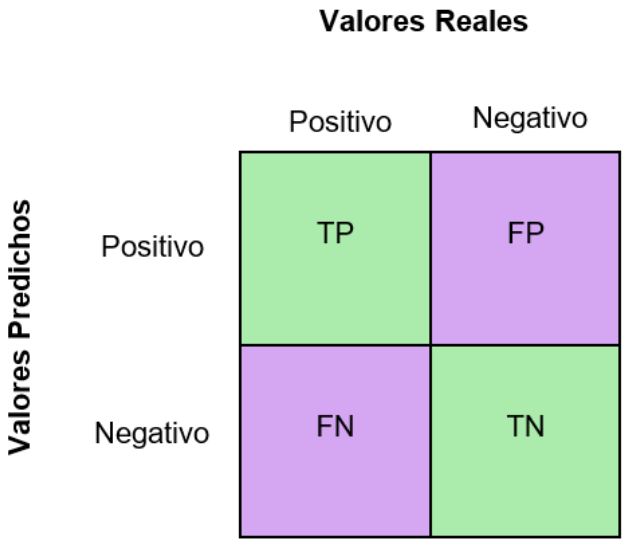
2.8.1. Matriz de confusión

De acuerdo con Géron (2017), la matriz de confusión es una herramienta que permite evaluar el rendimiento de un modelo de aprendizaje automático en un problema de clasificación. Consiste en una tabla que muestra la cantidad de predicciones correctas e incorrectas realizadas por el modelo para cada una de las clases objetivo.

Esta matriz es una representación de 2x2 que contiene los valores reales o actuales (verdadero o falso) y los valores predichos por el modelo (positivo o negativo). Estas combinaciones resultan en cuatro posibles resultados, los cuales se muestran en la Figura 1:

Figura 1

Matriz de Confusión



A continuación, se proporciona una descripción de las abreviaciones utilizadas en la matriz de confusión:

- **Verdaderos positivos (TP):** Representa el número de casos en los que el modelo clasificó correctamente una instancia como positiva.
- **Falsos positivos (FP):** Se refiere al número de instancias que fueron clasificadas incorrectamente como positivas por el modelo, aunque en realidad eran negativas.
- **Falsos negativos (FN):** Indica el número de instancias que fueron clasificadas incorrectamente como negativas por el modelo, a pesar de ser realmente positivas.
- **Verdaderos negativos (TN):** Indica el número de instancias que fueron clasificadas correctamente como negativas por el modelo.

Para este proyecto la matriz de confusión permitirá evaluar cómo el modelo está realizando las clasificaciones en cada clase (comentarios positivos, negativos y neutros) y determinar qué tipo de errores está cometiendo con mayor frecuencia. Esto ayudará a comprender el rendimiento

del modelo en términos de su capacidad para identificar y clasificar los sentimientos expresados en los tweets.

2.8.2. Precisión

Según Powers (2011), la precisión es una medida que indica la proporción de instancias clasificadas como positivas que fueron clasificadas correctamente. Esta métrica mide la capacidad del modelo para evitar clasificar erróneamente una instancia como positiva cuando en realidad no lo es. Su cálculo se realiza dividiendo el número de verdaderos positivos (TP) entre la suma de los verdaderos positivos y los falsos positivos (FP).

2.8.3. Exactitud

De acuerdo con Powers (2011), la exactitud indica la proporción de todas las instancias que fueron clasificadas correctamente por el modelo. Es decir, mide la capacidad general del modelo para hacer predicciones correctas, independientemente de la clase. Se calcula dividiendo el número de predicciones correctas (verdaderos positivos y verdaderos negativos) entre el número total de instancias.

2.8.4. Recuerdo (Exahustividad)

Según Powers (2011), esta métrica proporciona información sobre la capacidad del modelo para clasificar correctamente las instancias positivas. Su objetivo es medir la proporción de instancias positivas que fueron clasificadas correctamente. Para calcular esta métrica, se divide el número de verdaderos positivos (TP) entre la suma de los verdaderos positivos y los falsos negativos (FN).

2.8.5. F1-score (Puntuación F1)

En base a Powers (2011), esta métrica combina la precisión y la exhaustividad para evaluar el rendimiento de un modelo en problemas de clasificación. Proporciona una medida equilibrada entre la precisión y la exhaustividad, teniendo en cuenta tanto los falsos positivos como los falsos

negativos. Además, provee una medida global del rendimiento del modelo al considerar tanto los errores de clasificación positivos como negativos.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo

3.1. Investigación Cuantitativa

Según Hernández et al. (2014) la investigación cuantitativa se enfoca en la medición y análisis de datos numéricos, lo que permite la obtención de resultados precisos y la identificación de patrones y relaciones estadísticas. En el presente trabajo, se llevará a cabo una investigación cuantitativa para analizar los sentimientos expresados en los tweets de la cuenta de Pizza Hut Ecuador, con el objetivo de determinar la opinión de los usuarios sobre la empresa.

La recopilación de datos se llevará a cabo extrayendo información de la cuenta oficial de Twitter de Pizza Hut Ecuador. Después de esto, los datos serán preparados y etiquetados para poder ser procesados por el modelo de aprendizaje automático. Para analizar el rendimiento de los modelos utilizados, se emplearán métricas de evaluación objetivas mencionadas en el capítulo anterior. De esta manera, se podrá realizar un análisis exhaustivo y cuantitativo del desempeño de los modelos en base a estas métricas.

La aplicación de esta investigación en el proyecto permite obtener información precisa y objetiva acerca de la opinión de los usuarios de Twitter con respecto a Pizza Hut Ecuador. Los resultados obtenidos pueden ser de gran utilidad para la empresa, ya que pueden ser usados como referencia para mejorar sus productos y servicios y satisfacer de manera efectiva las necesidades de sus clientes.

Además, este tipo de proyecto puede resultar beneficioso tanto para entidades públicas como privadas que deseen conocer la opinión de los usuarios sobre un producto o servicio específico. La información recopilada a través de este estudio proporciona una perspectiva valiosa y basada

en datos reales que puede guiar la toma de decisiones y las estrategias de mejora en el ámbito empresarial.

3.2. Metodología CRISP-DM

La adopción de CRISP-DM como metodología para el proyecto de titulación proporciona una estructura clara y organizada. Su enfoque estructurado y sistemático brinda una guía paso a paso para el desarrollo de proyectos de minería de datos, lo que facilita la planificación, ejecución y control del proyecto en su totalidad. Al seguir esta metodología, se asegura un enfoque organizado y riguroso en cada etapa del proyecto, lo que contribuye a obtener resultados más consistentes y confiables.

Además, CRISP-DM permite una gestión efectiva del proyecto al proporcionar directrices claras y procesos bien definidos. Esto incluye la comprensión del negocio y la definición de los objetivos del proyecto, el manejo adecuado de los datos, la construcción y evaluación de modelos de aprendizaje automático, y la interpretación de los resultados obtenidos. Al seguir esta metodología, se maximizan las posibilidades de éxito en el logro de los objetivos establecidos en la tesis, ya que se cuenta con un marco de trabajo sólido y probado para abordar cada etapa del proyecto de manera eficiente.

4. Desarrollo

4.1. Extracción de la data

La extracción de los datos es una parte fundamental del proyecto, ya que la información recopilada es necesaria para realizar el análisis de sentimientos. Para el desarrollo del proyecto, se ha utilizado un Notebook con Python en el que se empleó la librería Snscrape para realizar la extracción de datos de la cuenta oficial de Twitter de Pizza Hut Ecuador.

Se realizó una consulta con parámetros específicos, la cual se puede observar en la Figura 2:

Figura 2

Código para la extracción de datos

```
!$HOME/.local/bin/snscrape --jsonl --progress --max-results 5000
twitter-search "to:PizzaHutEC" > $HOME/pizzahut-tweets.json
```

En la Figura 2 se visualiza como se define el límite de tweets que para este caso fue de 5000 y el criterio de búsqueda "to: PizzaHutEc", lo que permite obtener los comentarios o menciones dirigidas directamente a la cuenta oficial de Pizza Hut Ecuador en Twitter. Al utilizar este parámetro en la consulta, se obtiene un conjunto de comentarios relevantes ya que fueron realizados por los usuarios de la plataforma.

Como resultado se obtuvieron 4475 comentarios en formato json, los cuales fueron almacenados en un archivo llamado pizzahut-tweets.json

4.2. Limpieza y preparación de los datos

4.2.1. Limpieza de los datos

Los datos obtenidos y almacenados en el archivo .json contienen varias características, incluyendo:

- ID: Identificador único para cada tweet.
- Fecha: La fecha y hora en que se publicó el tweet.
- Contenido: El texto del tweet.
- Número de retweets: La cantidad de veces que el tweet ha sido retuiteado.
- Número de likes: El número de "Me gusta" que ha recibido el tweet.
- Información del usuario: Esta incluye detalles sobre la persona que realizó el comentario:
  - Nombre de usuario: El nombre de usuario en Twitter.
  - Número de seguidores: La cantidad de seguidores que tiene el usuario.
  - Número de personas seguidas: La cantidad de personas a las que el usuario sigue.
- Otros detalles adicionales en el conjunto de datos.

Estas características proporcionan información importante y detallada sobre cada tweet y su autor, lo que puede ser útil para análisis posteriores o para comprender mejor las interacciones en la plataforma de Twitter.

Para este caso en particular se va a utilizar únicamente el contenido del tweet, por lo tanto se guardan los comentarios en un archivo .csv llamado "tweets_pizzahut.csv". Este archivo contiene una columna que almacena el contenido de cada tweet y cada fila del archivo representa un tweet individual, como se muestra en la Figura 3.

Figura 3

Contenido de los tweets sin procesar

```
!head $HOME/tweets_pizzahut.csv
```

```
"@PizzaHutEC Para que si ya hace un año me hicieron lo mismo y nadie me dio respuesta de nada, conmig
o un cliente menos, y la falta de respuesta menos puntos para ustedes."
"@PizzaHutEC Si cuando se pide online pierden la orden y pedido y toca pedir reembolso del pago con t
arjeta, PÉSIMO SERVICIO! No pedir más aquí en la PH JIPIJAPA"
"@PizzaHutEC Pedí dos pizzas familiares + 1 cola Seven Up y me llegaron las dos pizzas pero con 1 Pep
si, y ya me habían facturado la otra cola y NO ME DAN SOLUCIÓN, pésimo."
"@PizzaHutEC señores este domingo no se que le paso a la app toda la noche tratado de hacer pedido pe
ro me dice que servicio a domicilio no hay"
"@PizzaHutEC no sirven sus canales de atención.  No sirve de nada llamar a hacer un pedido porque no
lo procesan #MuyMalServicio"
"@PizzaHutEC pensé que solo sucedió en el local del norte hacer menos de un mes denuncie lo mismo est
oy en el Quicentro norte y el precio de la factura es $4.99 sin embargo me cobran $5.00 sin dar cambi
o no es la cantidad es la actitud del personal. https://t.co/plI4KcL8G2"
"@PizzaHutEC por favor arreglen el aire acondicionado de su local de Salinas. Parecería que quieren t
ransmitir la experiencia de estar dentro del horno de pizza."
"@PizzaHutEC Más d media hora esperando para una pizza personal pésimo servicio en el @Mall_El_Forti
n"
"@PizzaHutEC fuimos a comer a Shopping de Playas en el patio de comidas y nos indicaron que comprando
2 pizzas familiares solo podían darnos 2 servilletas por pizza… es cierto esa nueva política????"
"@PizzaHutEC LES PARECE BIEN QUE MANDEN ESTA PORQUERIA DE PIZZA CON SUPUESTOS NACHOS CON QUESO QUE SO
N MIGAJAS Y 6 MISERABKES NUGGETS SIN EMPAFAR NI NADA... TODO FRIO Y MAS DE UNA DE ESPERA...  UNA PORQ
UERIA https://t.co/UjtUwsathP"
```

Para facilitar la limpieza de los datos, se importan los datos del archivo .csv a un DataFrame de la librería pandas. Además, se utiliza la librería re para trabajar con expresiones regulares, lo que nos permite realizar operaciones de limpieza y manipulación de los datos de manera más eficiente.

Este proceso implica cargar los datos del archivo .csv en un DataFrame utilizando la función adecuada de pandas, como read_csv(). Una vez que los datos se han cargado en el DataFrame, se utilizan las expresiones regulares adecuadas para la limpieza de los comentarios.

En la Figura 4 se puede observar que se crea una función limpiar_comentarios la cual toma cada comentario del dataframe como entrada y realiza varias operaciones de limpieza de texto basadas en expresiones regulares entre las que se encuentran: eliminar cualquier aparición de "RT" por los retweets, se eliminan URL, hashtag (#), menciones(@), eliminar caracteres especiales o no deseados que no sean letras, números o espacios en blanco y por último se convierte a minúsculas todo el texto de los comentarios.

Figura 4

Función para la limpieza de los comentarios

```
def limpiar_comentarios(texto):
    texto = re.sub(r'^RT[\s]+','', texto)
    texto = re.sub(r'https?:\/\/.*[\r\n*]','', texto)
    texto = re.sub(r'#','', texto)
    texto = re.sub(r'@[A-Za-z0-9_]+', '', texto)
    texto = re.sub(r'http\S+|www.\S+|pic.\S+|\S+.com\S*|\S+.org\S*|\S+.net\S*', '', texto)
    texto = re.sub(r'[^\w\s]', '', texto)
    texto = texto.lower().strip()
    return texto
```

```
df['comentarios_limpios'] = df['comentarios'].apply(limpiar_comentarios)
```

El resultado de la aplicación de la función “limpiar_comentarios” se asigna a una nueva columna llamada “comentarios_limpios” en el dataframe.

Posteriormente, se realiza una verificación para identificar la presencia de filas vacías, es decir, aquellas que no contienen información, y también se busca detectar la existencia de comentarios repetidos. Esta validación adicional es importante debido a la limpieza previa que se ha llevado a cabo en los datos, lo cual puede haber generado filas sin contenido.

El objetivo de esta verificación es garantizar la integridad de los datos y asegurar que se trabaje exclusivamente con filas que contengan información valiosa y única. Al eliminar las filas vacías y los comentarios repetidos, se evita cualquier sesgo o distorsión en el análisis posterior y se asegura que se esté trabajando con un conjunto de datos confiable y consistente.

En este caso, se identificaron un total de 141 filas vacías y 221 datos duplicados en el dataframe. Estos registros se han eliminado, lo que resulta en un conjunto final de 4113 comentarios válidos para el análisis.

4.3. Etiquetado de datos para el aprendizaje automático

Después de completar la etapa de limpieza de datos, se lleva a cabo el proceso de etiquetado, donde se asignan las categorías de positivo, negativo o neutro a cada comentario en base a su contenido. En este proyecto, se emplean dos métodos para realizar dicho etiquetado.

El primer método utilizado es el enfoque automático, donde se usa un algoritmo de clasificación de aprendizaje automático. Se entrena el modelo de Naive Bayes utilizando un archivo tipo json con un conjunto de datos previamente etiquetados y se aplica este modelo para predecir las categorías de los comentarios no etiquetados como se muestra en la Figura 5. Este enfoque automatizado es útil para procesar grandes volúmenes de datos y proporcionar una asignación rápida de etiquetas.

Figura 5

Modelo para predecir las etiquetas de los datos

```
from textblob.classifiers import NaiveBayesClassifier
def etiquetar_comentarios(texto, clasificador):
    return clasificador.classify(texto)
```

```
with open('/home/jupyter/diccionario.json', 'r') as fp:
    cl = NaiveBayesClassifier(fp, format='json')

df['etiquetas_NB'] = df['comentarios_limpios'].apply(etiquetar_comentarios, clasificador=cl)
```

Además de utilizar el enfoque de clasificación de etiquetas con aprendizaje automático, también se ha empleado la librería TextBlob para calcular la polaridad de los textos. Esta librería proporciona una medida de polaridad que va desde -1 hasta 1, donde un valor de 0 indica un sentimiento neutro, valores inferiores a 0 representan un sentimiento negativo y valores superiores a 0 indican un sentimiento positivo.

Por lo tanto, para aplicar la medida de polaridad con TextBlob, es necesario realizar la traducción de los comentarios al idioma inglés, ya que esta librería trabaja principalmente con ese idioma. Para llevar a cabo este proceso, se ha descargado el dataframe limpio como un archivo .csv y se ha utilizado la herramienta Google Translate para traducir los comentarios al inglés.

Una vez completada la traducción, se ha importado el nuevo dataframe que contiene los comentarios originales, los comentarios limpios, las etiquetas generadas por el modelo Naive

Bayes y la traducción de los comentarios. Esto permite aplicar la medida de polaridad de TextBlob y realizar un análisis más completo de los sentimientos expresados en los comentarios.

Cabe recalcar que la utilización de TextBlob ha permitido obtener una evaluación cuantitativa de la polaridad de los comentarios, lo cual brinda información adicional sobre la orientación positiva o negativa de los textos. Esto ha enriquecido el análisis de los datos al proporcionar una perspectiva objetiva y basada en métricas de sentimiento.

El segundo método utilizado es el enfoque manual, donde se revisan individualmente cada comentario y se determina la categoría correspondiente en función de su contenido y basándose en las etiquetas generadas con el enfoque automático. Este enfoque manual permite un análisis detallado de cada comentario, teniendo en cuenta los matices y el contexto para realizar una asignación precisa de las etiquetas.

La combinación de ambos métodos, manual y automático garantiza un etiquetado exhaustivo y preciso de los comentarios. Esto es fundamental para el análisis posterior y la interpretación de los datos en el proyecto de titulación. Al emplear estos métodos, se ha logrado un etiquetado confiable que sirve como base sólida para las etapas posteriores del proyecto.

4.4. Preparación de los datos

Después de haber etiquetado correctamente los datos, el siguiente paso consiste en transformar los comentarios en representaciones numéricas para que puedan ser procesados por los modelos de aprendizaje automático. Para lograr esto, se utiliza la técnica TF-IDF (Term Frequency-Inverse Document Frequency) que asigna un valor numérico a cada palabra en función de su frecuencia de aparición en un comentario y en el conjunto total de comentarios.

La transformación TF-IDF permite destacar las palabras más relevantes y discriminativas en cada comentario, ya que las palabras que aparecen con mayor frecuencia en un comentario y menos frecuencia en el conjunto total de comentarios tienen un mayor peso. De esta manera, se

obtiene una representación vectorial que refleja la importancia relativa de cada palabra en cada comentario.

Una vez que los comentarios han sido convertidos en vectores numéricos utilizando TF-IDF, se procede a realizar una tokenización utilizando BPE (Byte Pair Encoding). Esta técnica divide los vectores numéricos en subunidades más pequeñas llamadas tokens, lo que ayuda a capturar mejor las estructuras y patrones presentes en los textos. La tokenización mediante BPE permite una representación más eficiente y compacta de los comentarios, facilitando así el procesamiento posterior.

El siguiente paso consiste en realizar el mapeo de las etiquetas a valores numéricos. Este proceso es necesario ya que muchos algoritmos de aprendizaje automático requieren que las etiquetas sean representadas de forma numérica para su procesamiento.

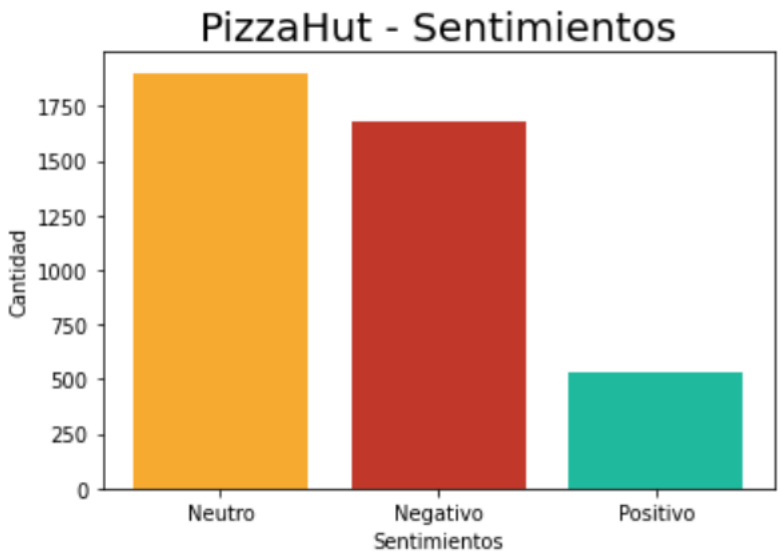
El mapeo de etiquetas a valores numéricos asigna un valor único a cada etiqueta, lo que permite que las etiquetas sean interpretadas por los algoritmos de aprendizaje automático como variables numéricas. Por ejemplo, si se tienen etiquetas de "positivo", "negativo" y "neutro", se puede asignar los valores numéricos 0, 1 y 2 respectivamente.

A continuación se realiza la separación de los datos en conjuntos de prueba y entrenamiento. Esta división es necesaria para evaluar y validar el rendimiento del modelo de aprendizaje automático.

Para este caso en particular, se utiliza una técnica de muestreo para equilibrar las clases en los conjuntos de datos de entrenamiento y prueba, ya que como se muestra en la Figura 6 existe un desequilibrio en la distribución de las clases, lo cual puede afectar el rendimiento y la precisión de los modelos de clasificación.

Figura 6

Distribución de las clases



Con el objetivo de abordar este desbalance, se realiza un proceso de sobremuestreo y submuestreo de los datos utilizando las librerías imblearn.over_sampling e imblearn.under_sampling.

El sobremuestreo consiste en aumentar la cantidad de instancias de la clase minoritaria, generando nuevas muestras sintéticas basadas en las instancias existentes. Esto permite equilibrar la distribución de las clases y proporcionar al modelo más información sobre la clase minoritaria.

Por otro lado, el submuestreo implica reducir la cantidad de instancias de la clase mayoritaria, eliminando aleatoriamente algunas de las instancias. Esto ayuda a evitar un sesgo hacia la clase mayoritaria y garantiza un equilibrio más adecuado entre las clases.

Al aplicar estas técnicas de ajuste de datos, se logra obtener un conjunto de datos balanceado en términos de las clases objetivo. Esto mejora la capacidad del modelo para capturar

correctamente las características de ambas clases y proporciona resultados más confiables y precisos.

Cabe recalcar que se revisaron aleatoriamente los datos que se añadieron en el sobremuestreo dado que es fundamental garantizar que los datos añadidos durante este proceso sean representativos y relevantes. Además, esto contribuye a la mejora del rendimiento del modelo y evita la introducción de ruido innecesario en el proceso de clasificación de sentimientos.

Por otro lado, es importante destacar que, previo al modelado, se realizó un análisis de la distribución de clases en los comentarios. Como se puede observar en la Figura 6, se evidencia una mayor cantidad de comentarios clasificados como negativos en comparación con los comentarios positivos.

Este hallazgo sugiere que existe una proporción significativa de clientes de Pizza Hut que expresan sentimientos negativos en relación con sus productos o servicios. Esta discrepancia entre comentarios negativos y positivos puede indicar que algunos clientes no están satisfechos con la calidad o la experiencia que han tenido al interactuar con la marca.

Es fundamental tener en cuenta esta información, ya que brinda una visión inicial sobre la percepción de los clientes y puede servir como una señal para la empresa de la necesidad de realizar mejoras en sus productos, servicios o en la atención al cliente.

4.5. Modelado de datos con aprendizaje automático

Una vez que los datos han pasado por la etapa de limpieza y preparación, se utilizan modelos de aprendizaje supervisado. Los modelos escogidos para este proyecto son: Random Forest, Support Vector Machines (SVM) y Naive Bayes. Estos modelos se seleccionan porque operan en base a un conjunto de datos previamente etiquetados, lo cual es fundamental para su funcionamiento y a su vez son especialmente adecuados para abordar tareas de clasificación.

En esta fase se utiliza la librería scikit-learn y en cada modelo se evalúa su rendimiento usando para ello la validación cruzada la cual que permite evaluar el desempeño del modelo utilizando múltiples divisiones del conjunto de datos en conjuntos de entrenamiento y prueba.

En lugar de entrenar y evaluar el modelo en una única división de datos, en este caso se realizan 5 divisiones y se promedian los resultados para obtener una estimación más robusta del rendimiento del modelo.

Además cabe mencionar que para todos los modelos se usa los datos tanto del sobremuestreo como del submuestreo para evaluar el rendimiento de los modelos y de esta manera poder determinar que modelo es el que brinda mejores resultados en términos de precisión.

4.5.1. Modelo Random Forest

Se procedió a entrenar un modelo RandomForestClassifier utilizando la librería scikit-learn y se evaluó su rendimiento. Tanto en la Figura 7 como en la Figura 8 se utiliza el mismo modelo solo que para el primero se usan los datos de sobremuestreo y para el segundo los de submuetsreo.

Figura 7

Código Random Forest Classifier con datos de sobremuestreo

RandomForestClassifier

```
from sklearn.ensemble import RandomForestClassifier

# Crear instancia del modelo
model = RandomForestClassifier()

# Realizar validación cruzada
scores = cross_val_score(model, X_over, y_over, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores)
print("Exactitud promedio:", np.mean(scores))
```

✓ 1m 23.4s

```
Exactitud de cada fold: [0.86471861 0.87987013 0.89057421 0.9176598  0.91982665]
Exactitud promedio: 0.8945298832622776
```

Figura 8

Código Random Forest Classifier con datos de submuestreo

RandomForestClassifier

```
# Realizar validación cruzada
scores = cross_val_score(model, X_under, y_under, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores)
print("Exactitud promedio:", np.mean(scores))
```

✓ 15.4s

```
Exactitud de cada fold: [0.77254902 0.7372549  0.73333333 0.70196078 0.74509804]
Exactitud promedio: 0.7380392156862745
```

4.5.2. Modelo SVM

Se procedió a entrenar un modelo SVM utilizando la librería scikit-learn y se evaluó su rendimiento. Como se puede apreciar en las MartFiguras 9 y 10, se utilizó el mismo modelo para ambos casos, sin embargo, en el primero se emplearon los datos del sobremuestreo y en el segundo se utilizaron los datos del submuestreo.

Figura 9

Código SVC con datos de sobremuestreo

SVC

```
from sklearn.svm import SVC

# Crear instancia del modelo
model = SVC()

# Realizar validación cruzada
scores = cross_val_score(model, X_over, y_over, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores)
print("Exactitud promedio:", np.mean(scores))
```

✓ 9m 45.6s

```
Exactitud de cada fold: [0.86904762 0.88203463 0.89057421 0.92524377 0.92741062]
Exactitud promedio: 0.8988621706931565
```

Figura 10

Código SVC con datos de submuestreo

SVC

```
# Realizar validación cruzada
scores = cross_val_score(model, X_under, y_under, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores)
print("Exactitud promedio:", np.mean(scores))
```

✓ 54.8s

```
Exactitud de cada fold: [0.75294118 0.74509804 0.7254902  0.69411765 0.75686275]
Exactitud promedio: 0.7349019607843138
```

4.5.3. Modelo Naive Bayes

Se llevó a cabo el entrenamiento de un modelo de Naive Bayes utilizando la variante BernoulliNB, dentro de la librería scikit-learn, y posteriormente se evaluó su rendimiento.

Como se puede apreciar en las Figuras 11 y 12, se utilizó el mismo modelo en ambos casos, sin embargo, en el primero se emplearon los datos del sobremuestreo, mientras que en el segundo se utilizaron los datos del submuestreo.

Figura 11

Código Naive Bayes con datos de sobremuestreo

Naive Bayes

```
from sklearn.naive_bayes import BernoulliNB

# Crear instancia del modelo Naive Bayes
model= BernoulliNB()

# Realizar validación cruzada
scores_nb = cross_val_score(model, X_over, y_over, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores_nb)
print("Exactitud promedio:", np.mean(scores_nb))
```

✓ 2.5s

```
Exactitud de cada fold: [0.81818182 0.81493506 0.81906826 0.85698808 0.82773564]
Exactitud promedio: 0.8273817731564211
```

Figura 12

Código Naive Bayes con datos de submuestreo

Naive Bayes

```
# Realizar validación cruzada
scores = cross_val_score(model, X_under, y_under, cv=5)

# Imprimir los resultados de la validación cruzada
print("Exactitud de cada fold:", scores)
print("Exactitud promedio:", np.mean(scores))
```

✓ 0.7s

```
Exactitud de cada fold: [0.75686275 0.76470588 0.7254902  0.71372549 0.72941176]
Exactitud promedio: 0.7380392156862745
```

4.6. Evaluación de los modelos

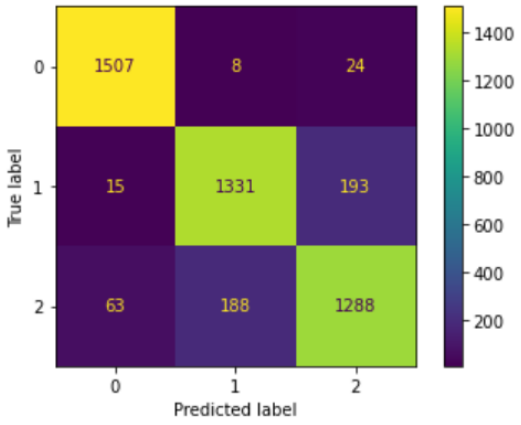
4.6.1. Modelo Random Forest Classifier

El modelo ha alcanzado una precisión promedio de 89.27% en el conjunto de datos sobremuestreados. Esto sugiere que el modelo es capaz de clasificar correctamente el 89% de los ejemplos en promedio en las tres categorías de sentimiento. El f1-score también es alto 89.30%, lo que indica un equilibrio entre precisión y recall.

A continuación en la Figura 13 se puede observar la matriz de confusión que detalla el número de predicciones de acertadas y fallidas del modelo.

Figura 13

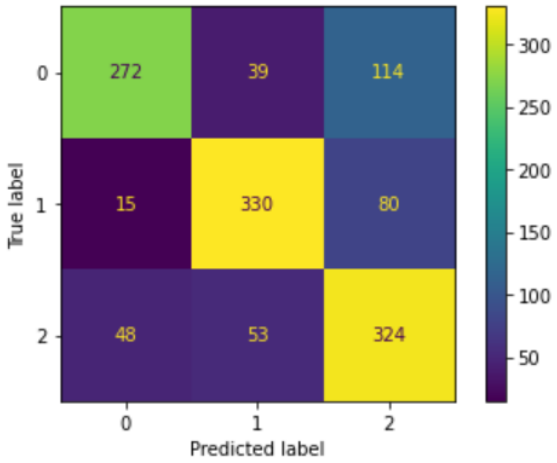
Matriz de confusión Modelo Random Forest con datos sobremuestreados



De igual manera al evaluar las métricas en el modelo con los datos submuestreados, se observa que tiene un rendimiento moderado, con una precisión promedio de aproximadamente 73.98%. En la Figura 14 se puede observar las predicciones del modelo en la matriz de confusión.

Figura 14

Matriz de confusión Modelo Random Forest con datos submuestreados



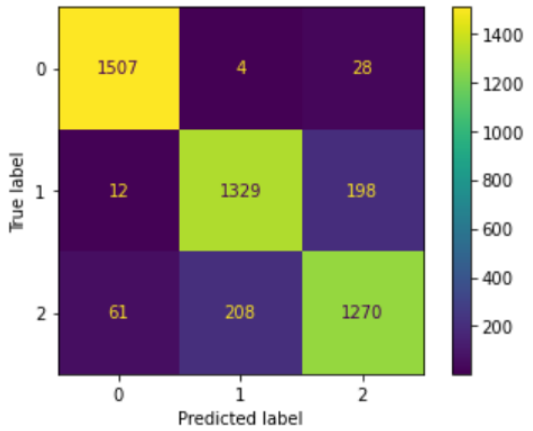
4.6.2. Modelo SVM

Para este modelo la precisión obtenida es de 89.92%, es bastante alta y refleja la capacidad del modelo para realizar predicciones precisas en el conjunto de datos con sobremuestreo. En otras palabras, cuando el modelo clasifica un comentario en una categoría de sentimiento (positivo, negativo o neutro), tiene una alta probabilidad de estar en lo correcto.

En la Figura 15 se puede apreciar la matriz de confusión la cual proporciona información detallada sobre las predicciones realizadas por el modelo.

Figura 15

Matriz de confusión Modelo SVM con datos sobremuestreados

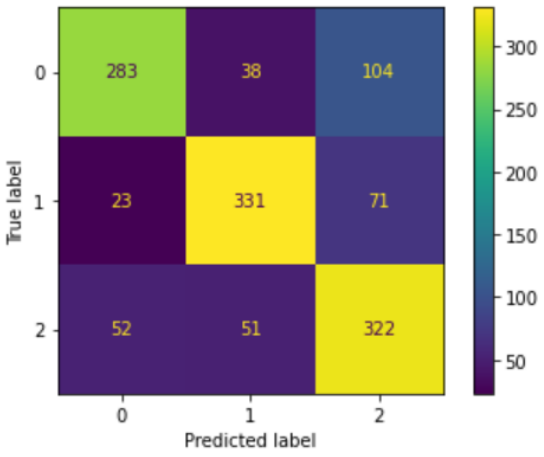


El mismo modelo SVM aplicado a los datos submuestreados ha demostrado una precisión de aproximadamente 74.98%. Aunque la precisión es ligeramente inferior a la obtenida con los datos sobremuestreados, sigue siendo un rendimiento moderado y muestra que el modelo es capaz de realizar predicciones precisas en la mayoría de los casos.

En la Figura 16, se muestra en detalle la matriz de confusión que representa las predicciones realizadas por el modelo.

Figura 16

Matriz de confusión Modelo SVM con datos submuestreados



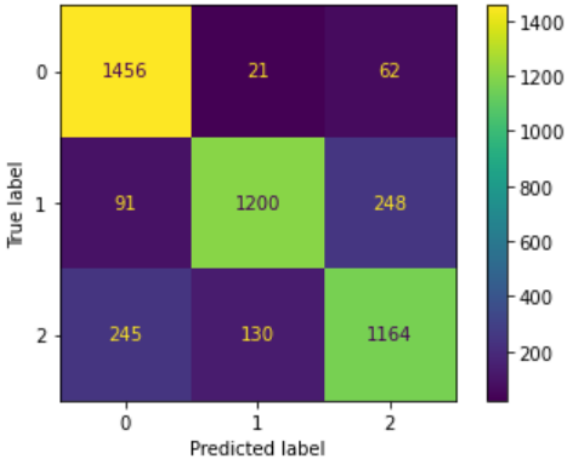
4.6.3. Modelo Naive Bayes

La precisión obtenida en el modelo de Naive Bayes con los datos sobremuestreados, con un valor de 0.8301, indica que aproximadamente el 83.01% de las predicciones realizadas por el modelo son correctas. Además, los puntajes de recall 82.74% y F1-score 82.58% también son altos, lo que sugiere que el modelo es capaz de recuperar correctamente la mayoría de las instancias.

A continuación en la Figura 17 se puede observar a detalle las predicciones de cada clase en la matriz de confusión.

Figura 17

Matriz de confusión Modelo Naive Bayes con datos sobremuestreados

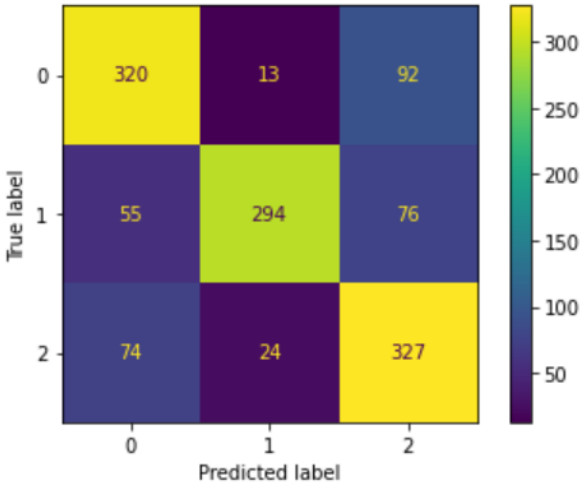


Al igual que con los otros modelos, al realizar el análisis del modelo de Naive Bayes con datos submuestreados, se obtuvo una precisión aproximada del 75.38%. Esta métrica indica la proporción de instancias clasificadas correctamente por el modelo.

En la Figura 18 se puede apreciar la matriz de confusión con cada una de las predicciones realizadas por el modelo.

Figura 18

Matriz de confusión Modelo Naive Bayes con datos submuestreados



4.6.4. Comparación de los modelos

Después de determinar la evaluación de los modelos por separado se realiza un análisis de los resultados obtenidos utilizando las métricas de evaluación previamente mencionadas: precisión, recall, F1-score y exactitud de cada uno de los modelos para poder determinar cuál es el modelo que mejor se adapta a los datos.

Tabla 1

Comparación de rendimiento de los modelos con datos sobremuestreados

Modelo	Precisión	Recall	F1-score	Exactitud
Random Forest	89.27%	89.36%	89.30%	89.45%
SVM	89.92%	89.88%	89.88%	89.89%
Naïve Bayes	83.01%	82.74%	82.58%	82.73%

Tabla 2

Comparación de rendimiento de los modelos con los datos submuestreados

Modelo	Precisión	Recall	F1-score	Exactitud
Random Forest	73.98%	72.63%	72.74%	73.80%
SVM	74.98%	73.49%	73.70%	73.49%
Naïve Bayes	75.38%	73.80%	74.03%	73.80%

Como se puede observar en la Tabla 1, los modelos de datos correspondientes al conjunto sobremuestreado muestran valores más altos en comparación con los modelos de datos submuestreados de la Tabla 2.

Ahora, al analizar los resultados obtenidos entre los modelos mencionados en la Tabla 1 se puede argumentar que los tres modelos muestran un rendimiento bastante similar en términos de precisión, recall y F1-score. Sin embargo, el modelo SVM es el más adecuado para este problema de clasificación de sentimientos, debido a que presenta un rendimiento sólido en términos de precisión, recall, F1-score y exactitud, superando a los otros dos modelos en la mayoría de las métricas evaluadas.

Además, el modelo SVM ha demostrado ser robusto en otros proyectos de clasificación y es ampliamente utilizado en la industria debido a su capacidad para manejar eficientemente conjuntos de datos de gran tamaño.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El uso de aprendizaje automático para el análisis de sentimientos en el contexto de una empresa de alimentos, como Pizza Hut Ecuador, ofrece importantes beneficios. Permite a la empresa comprender mejor la opinión y el sentimiento de los usuarios en relación con sus productos y servicios, identificar áreas de mejora y oportunidades de crecimiento, y ajustar su estrategia.
- La extracción de datos desde la cuenta de Twitter de Pizza Hut Ecuador utilizando Snscrape fue una elección acertada. Esta herramienta demostró ser eficiente y fácil de usar, permitiendo obtener los comentarios de los usuarios.
- La limpieza, etiquetado y preparación de los datos desempeñaron un papel crucial en el proyecto. La correcta manipulación de los datos recopilados permitió obtener resultados más precisos y confiables en los modelos de datos.
- El uso de modelos de aprendizaje automático supervisado, como Random Forest, SVM y Naive Bayes, fue fundamental para el análisis de sentimientos y mediante la aplicación de diversos algoritmos, se consiguió un modelo preciso.
- El modelo SVM destacó por su capacidad para realizar clasificaciones precisas, lo que demuestra su robustez y versatilidad, ya que pudo adaptarse adecuadamente a los datos y generar resultados satisfactorios sin requerir ajustes exhaustivos de los hiperparámetros.
- La aplicación de métricas de evaluación, como la matriz de confusión, la precisión, la exhaustividad, la exactitud y el F1-score, permitió medir de manera objetiva el desempeño de los modelos de análisis de sentimientos. Estas métricas brindaron una visión clara de la capacidad predictiva de los modelos y permitieron comparar su rendimiento en diferentes aspectos.

5.2. Recomendaciones

- Dado que los conjuntos de datos desbalanceados pueden afectar el rendimiento del modelo, se recomienda aplicar técnicas de balanceo, como submuestreo o sobremuestreo, para igualar la cantidad de instancias en cada clase de sentimiento. Esto mejorará la capacidad del modelo para aprender patrones de todas las categorías de sentimientos y evitará sesgos en la clasificación.
- La calidad de los datos es esencial para obtener resultados precisos, por lo que realizar una limpieza adecuada de los comentarios de los tweets, eliminando caracteres especiales, URLs, menciones de usuarios y cualquier otro ruido que pueda afectar la interpretación de los textos es fundamental.
- Si bien en este proyecto se utilizaron los hiperparámetros predeterminados de los modelos, se recomienda explorar técnicas de optimización de hiperparámetros, como la búsqueda exhaustiva o aleatoria, para encontrar la combinación óptima que maximice el rendimiento de los modelos.
- Para futuros proyectos se debería considerar la extracción de datos desde otras redes sociales como Facebook e Instagram, ya que de esta forma se ampliará las fuentes de información, permitiendo una visión más completa y precisa de las opiniones de los usuarios en diferentes plataformas.

BIBLIOGRAFÍA

Espin Riofrio, C., Cruz Chóez, A., & Zumba Gamboa, J. (2021). Métodos de extracción de comentarios de la red social Twitter para uso en Procesamiento de Lenguaje Natural. 63(11), 104–123. https://doi.org/10.23857/pc.v6i11.3257

Saura, J. R., Reyes-Menendez, A., & Palos-Sanchez, P. (2018). A feeling analysis in Twitter with machine learning: Capturing sentiment from #BlackFriday offers. Espacios, 39(42).

López Fierro, S., & Pacheco Villamar, R. (2021). Cuantificando opiniones expresadas en tweets durante las Elecciones Presidenciales Ecuatorianas del 2021 por medio del Análisis de Sentimiento. Ecuadorian Science Journal, 5(3), 209–219. https://doi.org/10.46480/esj.5.3.156

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1-135.

Hernández, R., Fernández, C., & Baptista, P. (2014). Metodología de la investigación (6th ed., pp. 4–21). Mcgraw Hill Education.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

Jurafsky, D., & Martin, J. H. (2019). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson.

Boyd, D., & Ellison, N. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), 210-230.

Medina, K. R. (2022, September 6). Estadísticas de la situación Digital en Ecuador 2021-2022. Branch Agencia. https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-en-ecuador-2021-2022/

Martinez Heras, J. (2020, September 18). Random Forest (Bosque Aleatorio): combinando árboles. IArtificial.net. https://www.iartificial.net/random-forest-bosque-aleatorio/#%C2%BFQue_es_un_Random_Forest

Cámara, E. M., Valdivia, M. M., & Urena, L. A. (2011). Análisis de Sentimientos. IV Jornadas TIMM Tratamiento de la Información Multilingüe y Multimodal 7 y 8 de abril de 2011, 61.

Mckinney, W. (2018). Python for data analysis : data wrangling with pandas, NumPy, and IPython (2nd ed.). O'reilly Media.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/mcse.2007.55

Goyvaerts, J., & Levithan, S. (2012). Regular Expressions Cookbook. "O'Reilly Media, Inc."

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. O'reilly.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michael, V. M., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12. https://doi.org/2825-2830

Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.

IBM. (2021, August 17). Conceptos básicos de ayuda de CRISP-DM. Www.ibm.com. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview