UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas

Enfoque plano-jerárquico basado en modelo de aprendizaje automático para la clasificación de productos de e-commerce

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Autor:

Harold Enrique Cotacallapa Mamani

Asesor:

Mg. Nemias Saboya Rios

Lima, noviembre de 2023

DECLARACIÓN JURADA DE ORIGINALIDAD DE TESIS

Yo Nemias Saboya Rios, docente de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **"ENFOQUE PLANO-JERÁRQUICO BASADO EN MODELO DE APRENDIZAJE AUTOMÁTICO PARA LA CLASIFICACIÓN DE PRODUCTOS DE E-COMMERCE"** del autor Harold Enrique Cotacallapa Mamani tiene un índice de similitud de 4% verificable en el informe del programa Turnitin, y fue realizada en la Universidad Peruana Unión bajo mi dirección.

En tal sentido asumo la responsabilidad que corresponde ante cualquier falsedad u omisión de los documentos como de la información aportada, firmo la presente declaración en la ciudad de Lima, a los 23 días del mes de noviembre del año 2023.

Mg. Nemias Saboya

Firmado digitalmente por Mg. Nemias Saboya
DN: C=PE, OU=FIA, O=UPeU, CN=Mg. Nemias Saboya, E=saboya@upeu.edu.pe
Razón: Soy el autor de este documento
Ubicación:
Fecha: 2023.11.30 14:48:29-05'00'
Foxit PDF Reader Versión: 12.1.0

Mg. Nemias Saboya Rios

ACTA DE SUSTENTACIÓN DE TESIS

En Lima, Ñaña, Villa Unión, a los **15** día(s) del mes de **noviembre** del año 2023 siendo **las 14:00 horas**, se reunieron en modalidad virtual u online sincrónica, bajo la dirección del Señor Presidente del jurado: **Dra. Erika Inés Acuña Salinas**, el secretario: **Ph.D. Javier Linkolk Lopez Gonzales**, y los demás miembros**: Mg. Danny Levano Rodríguez** y **el MSc. Fredy Abel Huanca Torres**, y el asesor, **Mg. Nemias Saboya Rios**, con el propósito de administrar el acto académico de sustentación de la tesis titulada: " Enfoque plano-jerárquico basado en modelo de aprendizaje automático para la clasificación de productos de e-commerce"

de el(los)/la(las) bachiller/es: a) **HAROLD ENRIQUE COTACALLAPA MAMANI**

…… b) **……**

conducente a la obtención del título profesional de **INGENIERO DE SISTEMAS**
(Nombre del Título profesional)

con mención en……

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): …… **HAROLD ENRIQUE COTACALLAPA MAMANI** ……

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
APROBADO	20	A	EXCELENCIA	EXCELENCIA

Candidato (b): …… **……** ……

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

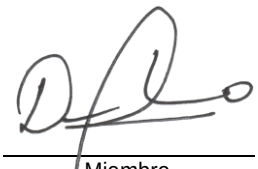
(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente
Dra. Erika Inés Acuña Salinas

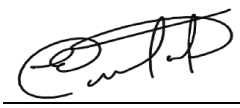
Secretario
PhD. Javier Linkolk Lopez Gonzales

Asesor
Mg. Nemias Saboya Rios



Miembro
Mg. Danny Lévano Rodríguez

Miembro
MSc. Fredy Abel Huanca Torres



Candidato/a (a)
Harold Enrique Cotacallapa Mamani

Candidato/a (b)

ÍNDICE

Enfoque plano-jerárquico basado en modelo de aprendizaje automático para la clasificación de productos de e-commerce

A flat-hierarchical approach based on machine learning model for e-commerce product classification

RESUMEN

En el ámbito del comercio electrónico, optimizar el proceso de clasificación de productos adquiere una importancia crucial debido a su influencia directa en la eficiencia operativa y, por ende, en la rentabilidad. Aunque se han dedicado esfuerzos académicos considerables para abordar este problema, persisten lagunas en la literatura existente. En tal sentido, este artículo presenta una solución para la clasificación jerárquica de productos de comercio electrónico usando un conjunto de datos de 4 niveles de profundidad, obtenidos de una destacada plataforma de comercio electrónico en América Latina. Nuestra propuesta consiste en un modelo de aprendizaje automático que integra enfoques tanto el clasificación plana como local (jerárquica) para mejorar la eficacia individual de cada uno. En busca de este objetivo, se llevó a cabo un análisis comparativo de siete algoritmos de aprendizaje automático: Multinomial Naive Bayes Multinomial, Linear Support Vector Classifier, Multinomial Logistic Regression, Random Forest, XGBoost, FastText y Voting Ensemble. Los tres primeros se utilizaron para el modelo que emplea el enfoque local, mientras que el modelo que usa el enfoque plano es el Voting Ensemble compuesto por los 3 primeros algoritmos mencionados anteriormente. Los resultados demostraron que este enfoque plano-jerárquico superó al mejor modelo de enfoque plano en un 0.15% y al mejor modelo de enfoque local (Clasificador Local por Nivel) en un 4.88%, medido por el puntaje F1 ponderado. Además, se pone a disposición un nuevo conjunto de datos en español con más de un millón de productos de comercio electrónico. Finalmente, se discuten las mejores técnicas de preprocesamiento para este conjunto de datos, junto con las limitaciones inherentes al estudio y las posibles direcciones para futuras investigaciones en esta área.

ABSTRACT

Within the e-commerce sphere, optimizing the product classification process assumes pivotal importance, owing to its direct influence on operational efficiency and, by extension, profitability. While extensive scholarly efforts have addressed this issue, persistent gaps remain within the existing literature. Therefore, this paper introduces a solution for hierarchical classification using a 4-level electronic product dataset obtained from a renowned e-commerce platform in Latin America. Our proposal consists of a Machine Learning model that integrates both flat and local (hierarchical) classification approaches to enhance each individual's efficacy. In pursuit of this goal, a comparative analysis of seven machine learning algorithms, including Multinomial Naive Bayes, Linear Support Vector Classifier, Multinomial Logistic Regression, Random Forest, XGBoost, FastText, and Voting Ensemble, was conducted. The first three were used for the model employing the local approach, while all seven were used for the model with the flat approach. The results demonstrated that this flat-hierarchical approach outperformed the best flat approach model by 0.15\% and the best local approach model (Local Classifier per Level) by 4.88\%, as measured by the weighted F1-score. Additionally, a new dataset in Spanish with over one million e-commerce products is made available. Finally, the best preprocessing techniques for this dataset are discussed, along with the study's inherent limitations and future research directions in this field.

I INTRODUCCIÓN

El auge de las plataformas de comercio electrónico en los últimos años y los desafíos que trajo la pandemia de COVID19, aceleró aún más la transformación digital de los países sub-desarrollados. Incluso después de la reapertura de las tiendas físicas, algunos países latinoamericanos mantienen un porcentaje de crecimiento en ventas por comercio electrónico por encima del promedio mundial (10.4%) durante el año 2023, esta lista es liderada por Brasil, seguido por Argentina y México, con el 17.0%, 14.0% y 13.5% respectivamente.

Al respecto, Gupta et al. [1] y Das et al. [2] afirman que el éxito de una plataforma de comercio electrónico depende significativamente del sistema de clasificación de productos, el cual consiste en el proceso de asignar una ruta de categoría a un producto, dentro de una estructura jerárquica [3]. Esta estructura se organiza desde las categorías más generales a categorías más específicas (e.g. Tecnología > Computación > Laptops y Accesorios > Laptops), lo cual contribuye a una rápida y precisa recuperación de los productos deseados por el cliente [4].

No obstante, el gran volumen de datos de las plataformas de comerio electrónico ha resaltado aún más los desafíos de la clasificación jerárquica de productos, tales como: a) la ambiguedad y variabilidad de las descripciones de los productos, estas pueden variar en estilo, estructura y tamaño, dificultando la extracción de características [5]; b) desbalanceo de datos, millones de productos distribuidos en categorías con un alto grado de dispersión y con una cola larga de distribución asimétrica [6]; c) multilingualismo, la presencia internacional de un marketplace exige tener soluciones en múltiples lenguajes [7], [8]; d) escalabilidad, la naturaleza digital de este modelo de negocio y los millones de productos de estas plataformas requiere un sistema de clasificación altamente efectivo y con resultados en tiempo real [9], [10].

Estudios recientes en el aprendizaje de máquina han permitido abordar con mejores herramientas los desafíos propios de la clasificación de productos, estas investigaciones abarcan el acceso a nuevos datasets públicos [3], [11], la aplicación de modelos pre entrenados basados en transformers [8], [12]–[14], el uso de modelos multimodales [15]–[17] y la aplicación de técnicas como machine translation [18] y transfer learning [7].

Por otra parte, los diversos modelos de clasificación jerárquica de la literatura pueden agruparse en 3 enfoques principales: clasificación plana, clasificación local, y clasificación global (big-bang) [19], cuyo rendimiento ha sido comparado y cuestionado en diversos estudios [5], [20], [21], sin embargo, los resultados de qué enfoque es mejor varían en el tiempo, esto se debe a las características de los datos de estudio, la técnica de representación de texto, el uso de algoritmos tradicionales o de aprendizaje profundo, entre otros factores; hasta donde se sabe, aún no se han explorado los resultados de unir ambos enfoques y aprovechar sus fortalezas para mejorar la clasificación de una estructura jerárquica.

Motivados por este vacío en la literatura, el objetivo principal del presente estudio es comparar el rendimiento de algoritmos de machine learning para la clasificación jerárquica de productos de comercio electrónico usando el enfoque de clasificación plana, local y una combinación de ambos.

Para alcanzar dicho objetivo, más de 1 millón de productos fueron recolectados de un reconocido marketplace en Latinoamérica, Mercado Libre Perú, utilizando su API pública (https://developers.mercadolibre.com.pe/), no obstante, este artículo usa solamente un subconjunto de los datos recopilados. El dataset completo consta de 31 categorías en el primer nivel y tanto el conjunto completo de datos como la porción utilizada en esta investigación son de acceso público a través de la plataforma Zenodo (https://doi.org/10.5281/zenodo.8415496).

Asimismo, se definieron 7 algoritmos de clasificación, agrupados en tres enfoques, los algoritmos tradicionales: Multinomial Naive Bayes (MNB), Multinomial Logistic Regression (MLR), Linear Support Vector Classifier (LSVC); algoritmos de aprendizaje ensamblado: XGBoost (XGB), Random Forest (RF), Voting ensemble (hard); y también un algoritmo basado en redes

neuronales: FastText (FT), con resultados satisfactorios en tareas de clasificación de texto [22], [23]. El estudio también brinda un análisis sobre el impacto de las técnicas de pre procesamiento y extracción de características en el rendimiento de los algoritmos. Por último, la comparación del desempeño de los algoritmos se realizó mediante la medición del accuracy, y las versiones ponderadas de f1-score, recall, y precision.

El artículo está organizado de la siguiente manera: En la sección I se realiza una breve introducción explicando el problema y los objetivos de la investigación, en la sección II se señala y describe brevemente los trabajos relacionados más importantes según la revisión de la literatura. En la sección III se describen aspectos teóricos sobre los enfoques de clasificación y los algoritmos utilizados. Los recursos y la metodología de investigación se detallan en la sección IV, los resultados son interpretados en la sección V y en la sección VI se discuten y clarifican algunos aspectos específicos de la sección anterior. Finalmente, las conclusiones del estudio se exponen en la sección VII.

II TRABAJOS RELACIONADOS

La variedad de atributos y el gran volúmen de datos de productos de e-commerce, ha motivado a los investigadores a abordar este problema desde distintas perspectivas. Uno de los primeros esfuerzos fue la construcción de estándares internacionales con el fin de unificar la estructura jerárquica de las distintas plataformas de comercio electrónico y facilitar su control [13], [24], [25], los más usados y conocidos son: The United Nations Standard Products and Services Code2 (UNSPSC) y el Global Product Classification3 (GPC), los cuales aún están vigentes.

En el contexto de machine learning, las investigaciones pueden clasificarse por el tipo de objeto de estudio: texto [12], [13], [26], [27], imagen [28]–[30] o ambos [16], [17], [31]; por el tipo de algoritmo usado: machine learning [25], [32], [33], deep learning [5], [13], [15], [34]; o según el tamaño del dataset: pequeño [31], [33], extenso [2], [11], [21]; así como por el tipo de enfoque de clasificación: flat, local, big-bang [19], [20], [35]. Es más, solamente entre las investigaciones que usan el texto como objeto de estudio, estas varían entre sí, ya que usan diversos atributos del producto de acuerdo a los datos disponibles, ellos pueden ser: título, descripción, precio, marca, breadcrumbs, entre otros; no obstante, la mayoría suele trabajar con el título y/o la descripción del producto [4], [8], [12], [13], [27]. A continuación, se describen algunas investigaciones que utilizan atributos textuales del producto.

Uno de los primeros sistemas que aplicó técnicas de recuperación de información y algoritmos de aprendizaje automático para abordar este problema, fue GoldenBullet, el cual alcanzó una precisión de 78% usando el algoritmo Naive Bayes con un enfoque plano y un dataset de 41 mil productos, los investigadores también desarrollaron modelos con un enfoque local, pero estos no superaron el enfoque plano [36]. Chavaltada et al. [37] contrastó el desempeño de múltiples algoritmos tradicionales de aprendizaje de máquina usando un enfoque de clasificación plana, donde Naive Bayes demostró tener los mejores resultados. Del mismo modo, Verma et al. [31] trabajó con 85 mil títulos e imágenes de productos para desarrollar el Multi-Modal Multi-level Boosted Fusion Learning Framework, que alcanzó un 90.53% en la versión macro de F1-score. Recientemente, Oancea [38] usó el título de los productos para comparar el rendimiento de 13 modelos de clasificación tradicionales, donde los algoritmos de Regresión Logística y Máquina de Sorporte Vectorial sobrepasaron a los otros según las métricas de precisión y f1-score ponderado.

En lo que concierne a datasets de gran escala, Walmart presentó el sistema Chimera como una solución escalable y precisa, combinando algoritmos de aprendizaje automático (Naive Bayes y Perceptron), reglas de decisión y crowdsourcing; este sistema usó 852 mil datos de entrenamiento organizados en 3663 clases de productos [39]. En estos casos es común aplicar algoritmos con arquitecturas basadas en redes neuronales y LSTM, por ejemplo, Ha et al.

[9] desarrolló el modelo DeepCN, basado en múltiples redes neuronales recurrentes y usando un dataset de 94 millones de productos; de la misma manera, Xia et al. [40] combinó mecanismos de atención y redes neuronales convolucionales para disminuir el tiempo de entrenamiento y evitar el uso de ingeniería de características durante la reducción de la dimensionalidad. Asimismo, en el SIGIR 2018 eCom Rakuten Data Challenge el modelo ganador fue un ensamble bidireccional de 6 LSTMs que alcanzó un f1-score ponderado de 85.13%, el dataset constaba de 1 millón de productos [34]. Sumado a esto, Das et al. [2] demostraron que al agregar el precio y la ruta de navegación del producto puede mejorar significativamente el rendimiento del clasificador, los autores usaron 2 datasets de gran escala, uno de Rakuten y otro de Amazon.

Por otra parte, en cuanto al tipo de enfoque utilizado para construir los modelos de clasificación, tanto el enfoque plano como el enfoque local son los más comunes, sin embargo, recientes investigaciones demuestran un especial interés en aprovechar mejor las relaciones jerárquicas entre las clases [12].

Entre los estudios más recientes que utilizan el enfoque plano en sus modelos de clasificación, podemos mencionar a Oancea [38], Hafez et al. [33] y Akritidis et al. [26] quienes utilizan algoritmos de Machine Learning (ML) tradicionales, mientras que, Ozyegen et al. [12], Lehmann et al. [7], Chen et al. [41], Skinner [34] y Suzuki et al. [42] usan algoritmos de Deep Learning; considerando solamente atributos textuales del producto para sus modelos.

Asimismo, dentro de los estudios que adoptan el enfoque de clasificación local, destacan Ozyegen et al. [12] y Brinkmann y Bizer [13], quienes hacen uso de modelos preentrenados basados en una arquitectura de transformers, el primero usa un clasificador local por nivel (LCL) y el segundo usa un clasificador local por nodo (LCN). Asimismo Allweyer et al. [25] demostraron un mejor rendimiento combinando el algoritmo Support Vector Machine (SVM) con el método TF-IDF usando el enfoque LCL; mientras que los resultados de Vandic et al. [35] favorecen la combinación de Naive Bayes con el método de Ganancia de Información (IG) usando el enfoque LCN, estos 2 últimos autores comparan el desempeño de diversos algoritmos tradicionales de ML, entre ellos: K-NN, SVM, Naive Bayes, Random Forest, Single Layer Perceptron, entre otros.
Ante la diversidad de estudios, algunos autores han intentado comparar qué tipo de enfoque de clasificación es mejor; para Krishnan et al. [5] el enfoque plano es mejor que el enfoque local según sus experimentos usando CNN y LSTM, mientras que Gao et al [21] asegura lo contrario, este propone un modelo con un enfoque LCL basado en Redes Neuronales que supera los clasificadores tradicionales planos (SVM, FastText, TextCNN) y jerárquicos (HSVM, HiNet); de hecho, Brinkmann y Bizer [13] también lograron resultados similares usando redes neuronales en su arquitectura.

No obstante, aunque aún no podemos asegurar con certeza qué tipo de enfoque de clasificación es mejor, sí podemos afirmar que la principal desventaja de un modelo con enfoque local recae en la inconsistencia taxonómica de la predicción [19], [21], en tanto que, la efectividad de un enfoque plano disminuye cuando existe un alto nivel de granularidad y los datos no están balanceados, ya que es más dificil para el modelo distinguir con claridad entre muchas categorías finales cuya similitud es más estrecha; pero, donde un enfoque es débil el otro enfoque es más robusto [19].

Por último, a pesar que este problema es antiguo existen pocos estudios que usan datasets cuyo idioma original es distinto al inglés. Entre ellos destaca el koreano [9], [24]; alemán [7], [25]; japonés [18], [40]; turco [12] y español [33]. De acuerdo a Liu et al. [11] este comportamiento se debe a que muchos datasets no son de acceso público. Además, cada lenguaje posee sus propias características y estas deben ser estudiadas particularmente [43]. Por ende, esta investigación también contribuye a la comunidad científica con un extenso dataset en español.

III. FUNDAMENTOS TEÓRICOS

A. ENFOQUES DE CLASIFICACIÓN JERÁRQUICA

Independientemente del dominio de aplicación, Silla y Freitas [19] abordan la clasificación jerárquica como un tipo de problema particular de la clasificación estructurada, donde la salida del algoritmo de clasificación es una taxonomía de clase, por ello, basándose en la forma en que el modelo explora la estructura jerárquica, [19] agrupa los modelos de clasificación en clasificadores planos, locales y globales (bigbang). Esta clasificación también es usada por otros autores [6], [13].

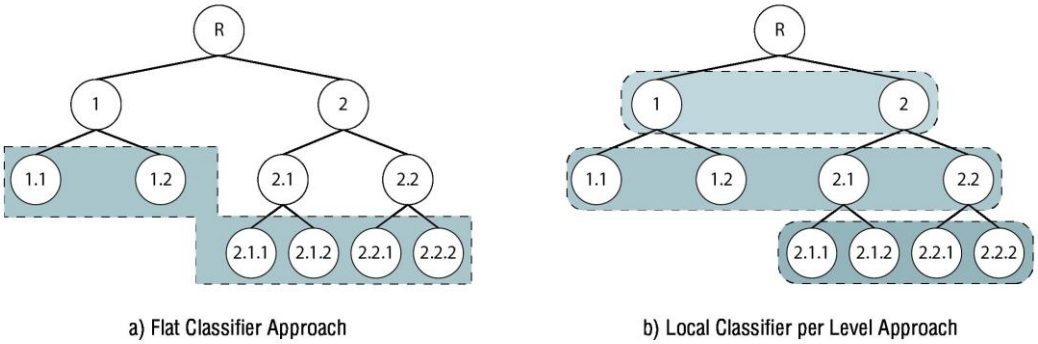


FIGURA 1: Principales enfoques de clasificación jerárquica

El **enfoque de clasificación plana** consiste en la predicción directa de las clases del último nivel (leaf nodes), e ignora la relación de padre-hijo entre las clases. No obstante, este enfoque resuelve indirectamente el problema de clasificación jerárquica, ya que, asumiendo que las clases de la jerarquía mantienen una relación "IS-A", al asignar una clase de nivel inferior a una instancia, también son asignadas, todas las clases predecesoras a esta, implícitamente [19]. La Figura 1 letra a. grafica este tipo de enfoque.

En contraste, el **enfoque de clasificación local** se subdivide en tres grupos, de acuerdo a cómo se utiliza la información local: Clasificador Local por Nodo Padre (LCPN), Clasificador Local por Nivel (LCL) y el Clasificador Local por Nodo (LCN). En el LCPN se entrena un clasificador multiclase por cada nodo padre, usualmente es el mismo algoritmo de clasificación. Para la construcción del dataset de entrenamiento se suele considerar el criterio de "siblings" y "exclusive siblings". Una ventaja notable es que usa menos clasificadores que el LCN [19]. Mientras tanto, el enfoque LCL consiste en entrenar un clasificador multiclase para cada nivel de la estructura jerárquica, y según el tipo de problema (single-label o multi-label) puede predecir una o varias clases en el respectivo nivel jerárquico [19]. Cuando se usa este enfoque es necesario complementar con un método de inconsistencia taxonómica [20]. El LCN consiste en entrenar un clasificador binario para cada nodo de la jerarquía, siendo muy ventajoso para un problema multi-label; no obstante, para [20] una desventaja notable es el número de clasificadores que puede alcanzar cuando se trabaja con datasets de gran escala. En la Figura 1 letra b. se visualiza el tipo enfoque utilizado en esta investigación.

Por último, el **enfoque de clasificación global (big-bang)** se basa en entrenar un único clasificador para todas las clases de la estructura jerárquica, considerando la jerarquía de clases como un todo. Este modelo es normalmente más pequeño que el tamaño total de todos los tipos de clasificadores locales, aunque carece de modularidad para el entrenamiento [19].

B. ALGORITMOS DE APRENDIZAJE DE MÁQUINA

1. Multinomial Naive Bayes

El clasificador multinomial Naive Bayes (NB) es una variante del algoritmo probabilístico NB usado para datos distribuidos multinomialamente, es decir, cuando se tienen múltiples clases y se desea modelar la probabilidad de que un evento pertenezca a cada una de estas [44]. Este modelo se basa en el teorema de Bayes y asume una independencia condicional entre las características cuando se conoce la clase, lo cual simplifica el cálculo de probabilidades y hace que el algoritmo sea computacionalmente eficiente [37].

2. Multinomial Logistic Regression

Se basa en el algoritmo probabilístico Logistic Regression (LR) [45], el cual es un clasificador lineal que busca encontrar una relación entre las características y la variable dependiente (clase) para predecir la probabilidad de que un ejemplo pertenezca a una clase específica. La variación multinomial [46] de este algoritmo utiliza una fórmula que suele incluir la función softmax o crea un conjunto de múltiples clasificadores binarios en un esquema one-vs-rest (OVR) [23].

3. Support Vector Machine (SVM)

SVM [47] es un algoritmo de aprendizaje supervisado no probabilístico, aplicado usualmente a problemas de clasificación binarios, pero, cuando se aplica a un problema de clases múltiples, divide internamente la tarea en múltiples problemas binarios y los resuelve usando muchos SVMs [48]. Para una clasificación lineal emplea vectores de soporte de cada clase para construir hiperplanos entre sí, mientras que, para una clasificación no lineal aplica una función kernel, la cual mapea los datos de entrada a un espacio de características de mayor dimensión [37]. En esta investigación, se utiliza el clasificador LinearSVC [49] cuyo kernel es lineal y es más eficiente para grandes conjuntos de datos 4 , además ha demostrado un buen rendimiento en problemas de clasificación multiclase [12], [27], [48], [50].

4. Random Forest (RF)

Introducido por Ho [51] y desarrollado por [52]. RF es un algoritmo basado en el aprendizaje ensamblado y usa el concepto de "bagging" para la selección de muestras. Combina múltiples árboles de decisión individuales para obtener predicciones más precisas y robustas [23]. Cada árbol de decisión es entrenado por un subconjunto del texto de entrenamiento, y en cada nodo del árbol se selecciona aleatoriamente un subconjunto de características; para la predicción, RF toma el voto mayoritario de todos los árboles individuales. No obstante, entrenar una gran cantidad de árboles puede ser costoso computacionalmente, requerir mayor tiempo de entrenamiento y usar mucha memoria [53].

5. eXtreme Gradient Boosting (XGBoost)

XGBoost [54], también es un algoritmo de aprendizaje ensamblado y se basa en la técnica Boosting para entrenar los clasificadores débiles y combinarlos en un modelo fuerte [33]. XGBoost construye árboles de decisión secuenciales y los agrega de manera iterativa, cada árbol ajusta los pesos dando mayor importancia a las instancias mal clasificadas y menor importancia a las correctas, de este modo el siguiente clasificador se enfocará en las instancias "difíciles" clasificadas por el modelo anterior [23]. Cada clasificador débil aprende utilizando una función objetivo compuesta por la función de pérdida y la función de regularización en cada iteración [55].

6. FastText

Esta librería fue creada por Facebook para la clasificación y aprender representaciones de palabras [22], [43]. Esta técnica se basa en la arquitectura skip-gram y a diferencia de las técnicas anteriores toma en cuenta la morfología de las palabras, logrando así una mejor representación vectorial para las palabras OOV [23]. Facebook disponibilizó modelos pre-entrenados para 294 idiomas diferentes, estos fueron entrenados en Wikipedia usando FastText con 300 dimensiones y el modelo skip-gram de Word2Vec con sus parámetros predeterminados [53].

7. Ensemble

Este tipo de algoritmos combina la predicción de diversos algoritmos base para mejorar la generalización o robustez de un modelo individual. Se pueden distinguir dos familias de métodos de ensamble: Average y Boosting, cuya principal diferencia está en cómo construyen los algoritmos base, el primero construye múltiples algoritmos base independientes y luego promedio sus predicciones, mientras que el segundo construye los algoritmos base secuencialmente buscando reducir el error en cada iteración [56]. También resuelven problemas de sobreajuste y obtienen buenos resultados con pocos datos de entrenamiento. [57] Este artículo usó el método de votación mayoritaria (hard voting), el cual consiste en clasificar una instancia según la clase que recibió mayor número de votos [58]. Otros estudios también han considerado este método dentro de sus experimentos [14], [15], [17], [28].

IV. MATERIALES Y MÉTODOS

La ejecución de esta investigación siguió los pasos ilustrados en la Figura 2, los cuales están basados en el conocido CRISP-DM [59]. En primer lugar, se recolectaron los datos de la plataforma de comercio electrónico y se realizó un análisis de datos exploratorio. En segundo lugar, se llevó a cabo la limpieza de los datos, y transformación de los mismos. Luego, los datos transformados se usaron para entrenar los modelos y hallar los mejores hiperparámetros. Por útltimo, se realizó una evaluación del rendimiento de cada algoritmo usando métricas para una clasificación multiclase. Cada una de estas etapas y cómo se aplican los distintos experimentos se explican en el resto de esta sección.

Los experimentos y el análisis de datos fueron ejecutados en un computador con las siguientes características: Sistema Operativo Linux 6.2.0-31-generic x86_64 con 64 núcleos de CPU, 128 GB de RAM y una tarjeta de video dedicada NVIDIA GeForce RTX 3080.

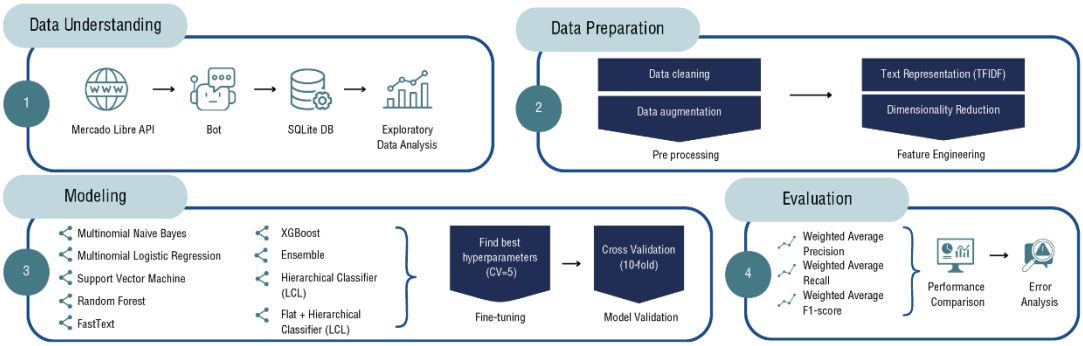


FIGURA 2: Metodología de Investigación basada en CRISP-DM.

A. DATOS

La Figura 3 ilustra el proceso de recopilación de datos y selección de muestras utilizado en esta investigación. Esta recopilación de datos se realizó entre febrero de 2022 y mayo de

2023, donde se utilizó un bot de Python para acceder a la API de Mercado Libre. Recopiló el código de producto, la categoría, el título, el precio, la unidad monetaria y el enlace del producto, almacenando esta información en una base de datos local de SQLite. En total, se recopilaron 1,198,398 productos únicos distribuidos en 31 categorías en el nivel superior. Este conjunto de datos, compuesto por los datos originales completos, se denominará Do de aquí en adelante.

Posteriormente, solo se seleccionaron las categorías relacionadas con productos tecnológicos, ya que son el grupo predominante en el conjunto de datos. Estas categorías son: Computación, Electrónica, audio y video, Celulares y teléfonos, Cámaras y accesorios y Consoloas y videojuegos, en total 303,508 productos. Además, se descartaron los niveles 5 y 6 de la estructura jerárquica original porque solo el 7,85% de los productos pertenecen a una clase en uno o ambos de estos niveles. Finalmente, se extrajo una muestra aleatoria estratificada de 170,332 instancias para reducir la complejidad del modelo.

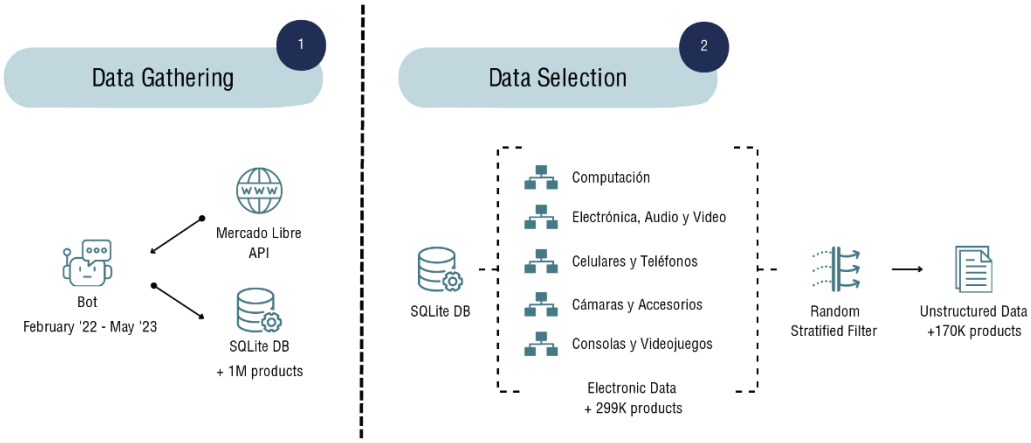


FIGURA 3: Metodología de Investigación basada en CRISP-DM.

Para asegurar la calidad de los datos y disminuir el ruido, primero todos los títulos fueron transformados en minúsculas y luego se aplicaron los siguientes métodos: 1) las clases con 1 instancia fueron removidas; 2) se eliminaron las categorias denominadas como 'Otros', [60]; 3) los títulos falsos fueron removidos; 4) se eliminó el código de Amazon Standard Identification Number (e.g 'b09mcz6ndg') y el número telefónico peruano que algunos productos poseían al final del título; 5) los títulos duplicados fueron eliminados junto con las celdas nulas. Al término de esta limpieza el dataset finalizó con 145,219 productos y este fue usado para todos los experimentos de esta investigación, al cual denotaremos como Ds de aquí en adelante.

TABLA 1: Descripción de variables del dataset

Variable	Data Type	Total	Example
title	categorical nominal	145,219	Dragon Touch Instant Print Kids Camera Instantf...b08hyn6zl6
text	categorical nominal	145,219	dragon touch instant print kids camera instantf...
taxonomy	categorical nominal	543	1039_430361_1040

La descripción de las variables de Ds se observan en la Tabla 1. La variable title es el título original del producto; text, es el título procesado con los métodos mencionados

previamente y taxonomy es la ruta de categorías a la que pertenece el producto, cada categoría tiene un código y está separada por un guión bajo, el primer código es la categoría mas general y el último es la categoría más específica.

Según el análisis exploratorio ejecutado, el conjunto de datos Ds tiene una estructura de árbol (Y), es Single Label Path (Ψ), y es Full Depth Labeling (Φ), es decir, las clases tienen únicamente un nodo raíz (Tecnología), cada nodo (clase) tiene un único nodo padre y todos los nodos tienen una etiqueta, por tanto, cada predicción debe realizarse hasta llegar a un nodo final. Esta nomenclatura fue propuesta por Silla y Freitas [19] para problemas de clasificación jerárquica.

Además, la Tabla 2 revela que a medida que aumenta el nivel jerárquico, el número de ejemplos disminuye y el número de clases aumenta significativamente. Esto implica que hay menos datos disponibles para que el modelo aprenda a discriminar entre las clases finales, un fenómeno que a menudo se denomina desafío de granularidad o de grano fino [11].

TABLA 2: Distribución de los datos por nivel jerárquico

	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Number of products	145,219	145,219	137,554	71,096
Number of classes	5	57	298	299
Number of products in leaf categories	0	7,665	66,458	71,096

Otras características de este dataset (Ds) también se observan en la Tabla 3. En el nivel 3, por ejemplo, existen 298 clases, las cuales tienen como mínimo 2 y máximo 6,522 productos y la mediana es igual a 97.5, lo cual representa una distribución con sesgo a la derecha; además, en promedio cada categoría posee 461.59 productos y la desviación estándar es igual a 811.52, casi el doble del promedio, lo que denota una alta variabilidad en los datos. Estas características no solo aplican para el nivel 3 sino para el dataset completo, lo cual también se coomprueba en la Figura 4.

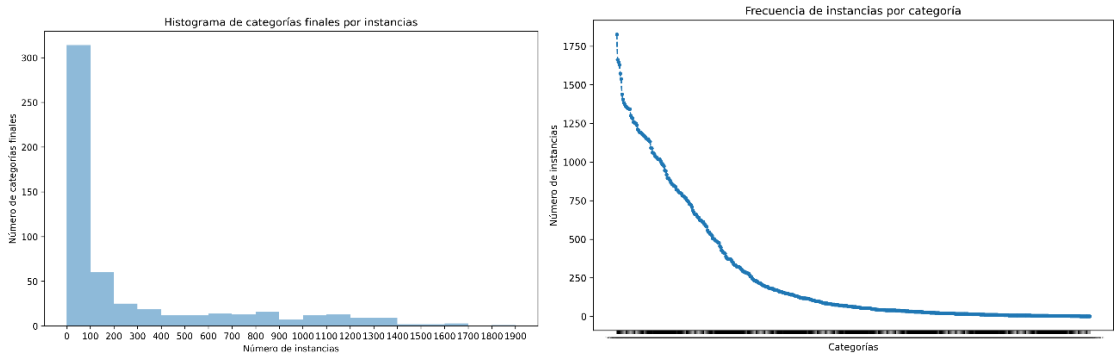


FIGURA 4: Distribución de los datos

Asimismo, las estadísticas descriptivas del título de los productos se muestran en la Tabla 4, estas indican que un título puede tener entre 2 o 143 caracteres, en promedio un título tiene 40.90 caracteres; cada título tiene en promedio 8 palabras, y esta palabra tiene 5.31 caracteres en promedio. En general, la dispersión en estos indicadores es bajo, sin embargo existen valores atípicos, por ejemplo, algunos títulos tienen 60 caracteres en un 1 token, debido a que todas las palabras del título están unidas por un '-' (e.g canon-r6- mirrorless-camera-rf-24-105mm-adapter-bag-flash-tri).

TABLA 3: Estadística descriptiva de la distribución de las instancias por categoría y por nivel

	Count	Min	Q25	Median	Q75	Max	Mean	Std
Nivel 1	5	7,719	18,746	26,613	26,729	65,412	29,043.8	21,764.10
Nivel 2	57	6	430	1,245	3,212	15,895	2,547.70	3,486.57
Nivel 3	298	2	18.5	97.5	661.5	6,522	461.59	811.52
Nivel 4	299	2	15	56	237.5	1,827	237.78	369.91
Leaf categories	543	2	15.5	63	333	1,827	267.44	397.23

TABLA 4: Estadística descriptiva del título del producto

	Min	Q25	Mediana	Q75	Max	Mean	Std
# chars per title	2	35	44	49	143	41.00	10.19
# tokens per title	1	6	8	10	29	8.00	2.42
Average length per token	1.5	4.5	5.13	6	60	5.32	1.35
# letters per title	1	29	37	43	137	35.56	9.74
# symbols per title	0	2	4	8	42	5.44	4.81

Para llevar a cabo el modelamiento, el dataset Ds fue dividido en un conjunto de entrenamiento y otro de pruebas siguiendo una proporción de 80 a 20, en la Tabla 5 se presenta la distribución de los datos para cada uno de los subconjuntos respectivamente.

TABLA 5: Distribución de los datos del dataset de entrenamiento y pruebas

	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Training data	115950	115950	109822	56754
Testing data	29269	29269	27732	14342

B. PRE-PROCESAMIENTO

Durante el pre procesamiento de los datos se codificaron 8 técnicas de limpieza y creación de características, y se realizaron múltiples experimentos con ellas para determinar cuál es la mejor combinación para cada algoritmo, las técnicas usadas fueron: remoción de los caracteres numéricos, reemplazo de caracteres no alfanuméricos (,.%#), reemplazo de patrones en el texto (medidas, abreviaciones, códigos), remoción de stop words y palabras con un carácter, stemming, lemmatization, función de n-grams personalizada, bi-grams y tri-grams. La aplicación de estas técnicas se describe en el pseudocódigo del Algoritmo 1.

Algorithm 1 Data Cleaning

Input: $titles$: titles to be processed
Output: $titles_p$: titles processed

procedure PREPROCESS($titles$)
  $n \leftarrow$ length($titles$)
  $titles_p \leftarrow []$
  for $i \leftarrow 1$ to n **do**
    $title_p \leftarrow$ remove_numeric_token($titles[i]$)
    $title_p \leftarrow$ replace_symbols($title_p$)
    $title_p \leftarrow$ replace_patterns($title_p$)
    $title_p \leftarrow$ remove_stopw_smallw($title_p$)
    $title_p \leftarrow$ stemming($title_p$)
    $title_p \leftarrow$ lemmatization($title_p$)
    $title_p \leftarrow$ custom_ngrams($title_p$)
    $title_p \leftarrow$ bigrams($title_p$)
    $title_p \leftarrow$ trigrams($title_p$)
    $titles_p \leftarrow titles_p$ + [$title_p$]
  end for
  $titles_p \leftarrow$ remove_empty_titles($titles_p$)
  return $titles_p$
end procedure

Para abordar el problema de la distribución sesgada o también conocido como long tail, esta investigación realizó el aumento de datos por categorías, tomando como referencia el método usado por Suzuki et al. [42]. Dicho método consiste en combinar todas las palabras de una categoría final en un arreglo y según el promedio de palabras del título se seleccionan aleatoriamente las palabras que conformarán un nuevo título en la categoría analizada. Para aplicar este método, se definió el límite θ, el cual denota la cantidad mínima de productos que todas las categorías finales deben tener, por consiguiente, el número de títulos nuevos generados es la diferencia entre el límite θ y el número de títulos existentes en la categoría; se consideraron 3 valores distintos para θ: 20, 60 y 100

C. INGENIERÍA DE CARACTERÍSTICAS

1. Representación de texto

Para la representación del texto en un espacio vectorial se utilizaron algunas técnicas que disponibiliza la librería scikitlearn: CountVectorizer y TfidfVectorizer; la primera calcula la frecuencia de palabras en un documento (TF), y la segunda técnica combina la primera con la frecuencia con que una palabra aparece en todos los documentos de la colección (TFIDF) con el fin de reducir el efecto de las palabras comúnes [53]. A partir de estos 2 métodos, se definieron 3 experimentos que fueron usados en la etapa de preparación de los datos: 1) CountVectorizer, 2) TF-IDF, 3) TF-IDF sublineal.

2. Reducción de Dimensionalidad

Tanto el método de Frecuencia de Términos (TF) como la combinación de este con la Frecuencia de Documentos Invertida (IDF) se basan en el vocubulario del corpus, es decir, la representación vectorial está sujeta a un número fijo de palabras únicas. En consecuencia, cuando se trabaja con un gran volumen de datos, el vocabulario se torna muy extenso y por ende el número de dimensiones de la matriz dispersa es elevado, lo cual requiere mayor capacidad de procesamiento y tiempo de entrenamiento de los modelos.

Por esta razón, existen técnicas de reducción de características, las más usadas y adaptadas para matrices dispersas son: el método de TSVD (Truncated Singular Value Decomposition), es rápido y escalable; LSA (Latent Semantic Analysis) también basado en la descomposición de valores singulares (SVD); y el LDA (Latent Dirichlet Allocation) que es un modelo prababilístico. El presente estudio implementa el método TSVD para la reducción de características considerando los valores de 25, 100, 500 y 1000 dimensiones.

D. MODELAMIENTO: ENFOQUE PLANO-JERÁRQUICO

La etapa de modelamiento de los algoritmos está compuesto por 2 pasos principales, como se visualiza en la selección del modelo de la Figura 5. El primer paso consiste en la búsqueda de los mejores hiperparámetros, tanto en la vectorización como en el mismo algoritmo de ML, para lo cual se usa el método GridSearchCV de Scikit-learn que emplea como estrategia de evaluación la técnica de cross-validation estratificado con 5-folds; el segundo paso es la validación de los resultados, para lo cual se utiliza 10-folds construidos a partir del dataset Ds, para cada fold se mezclaron aleatoriamente todos los datos, luego se extrajo de modo estratificado y aleatorio el 20% para el conjunto de pruebas y lo restante para el dataset de entrenamiento. El promedio de las métricas de rendimiento obtenidas de cada fold se observa en la Tabla 7 según el modelo.

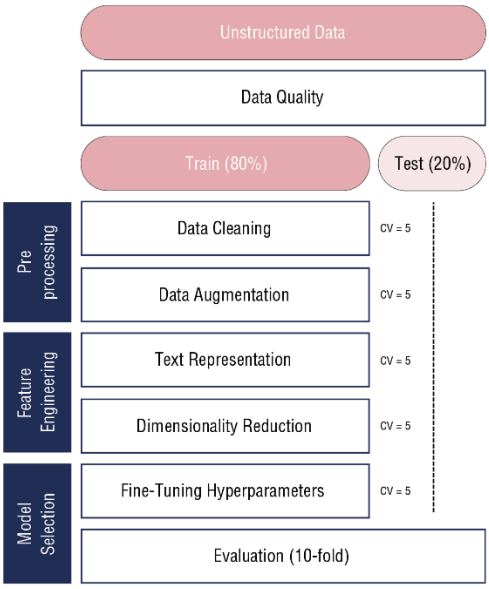


FIGURA 5: Selección del modelo con enfoque plano.

Respecto al modelo jerárquico que usa el enfoque de clasificación local por nivel (LCL), se eligieron los 3 mejores modelos no ensamblados del enfoque plano y fueron probados en cada nivel de la estructura jerárquica, como se ilustra en la Figura 6. Para el entrenamiento y validación de estos algoritmos en cada nivel se usó un conjunto de 5-folds obtenidos del conjunto 10-folds construido previamente. Según el nivel de la jerarquía, cada algoritmo fue entrenado usando todos los productos, pero, aquellos que no tenían una clase en dicho nivel recibieron una etiqueta por default igual a ’0’. En la Figura 6 se visualiza al lado izquierdo el proceso de construcción del modelo jerárquico. Los resultados de cada modelo por nivel se muestran en la Tabla 8.

Luego de elegir el mejor modelo por nivel, se construyó el modelo jerárquico (LCL) y se validó su rendimiento usando los mismos 10-folds que fueron usados para los modelos con enfoque plano. Sumado a esto, se realizaron 2 experimentos en la forma de uso del dataset de entrenamiento, en el primer experimento (hier-approach-1) se consideró como datos de entrenamiento solamente los productos que pertenecían a una categoría en el nivel de estudio, mientras que en el segundo experimento (hier-approach-2) se agregó una etiqueta igual a '0' para los productos que no tenían una categoría en algún nivel, de esta manera, todos los productos tenían un nivel de profundidad igual a 4, es decir, si la taxonomía de un producto era: '1500_850_26', con el experimento hierapproach-2 su nueva taxonomía sería: '1500_850_26_0'. Por último, la predicción final del algoritmo jerárquico (LCL) es la unión de la predicción individual de cada algoritmo, una por cada nivel, como se visualiza en la Figura 7.

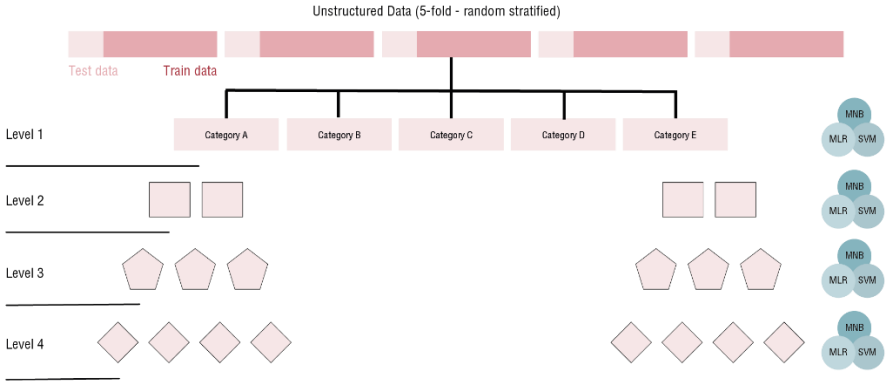


FIGURA 6: Selección del modelo con enfoque jerárquico (LCL).

Con el fin de evitar la inconsistencia predictiva del modelo jerárquico, aprovechar la clasificación modular del mismo y la generalización del modelo con enfoque plano, se construyó un modelo híbrido que combine ambos enfoques, como se visualiza en la Figura 7, ambos enfoques son combinados en la fase predictiva.

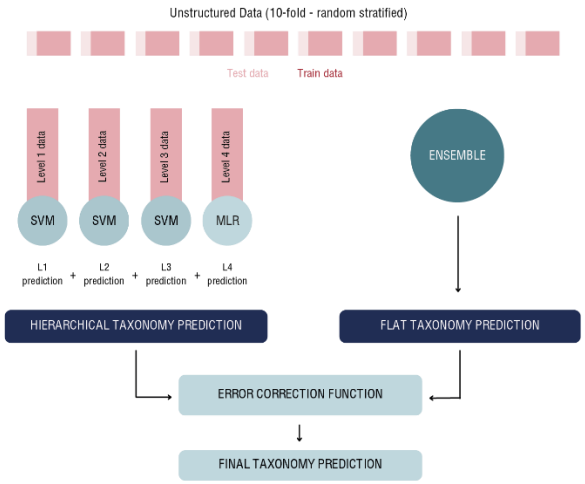


FIGURA 7: Enfoque Plano-Jerárquico

La combinación de ambos modelos sigue la lógica del Algoritmo 2, el cual analiza cada predicción del modelo jerárquico, y en caso no exista en la lista de taxonomías, entonces se

reemplaza por la predicción del modelo plano, en consecuencia, esta función evita la inconsistencia predictiva y conserva las predicciones reales del clasificador jerárquico.

Algorithm 2 Best Prediction Selection

Input: row: DataFrame Row
Input: $taxons$: List of all taxonomies
Output: $best\_prediction$: Best prediction for the row

```
procedure CREATEBESTPREDICTION(row, taxons)
    if row[taxo_joined] not in taxo_names then
        best_prediction ← row[taxo_ensemb]
    else
        best_prediction ← row[taxo_joined]
    end if
    return best_prediction
end procedure
```

E. EVALUACIÓN

Las métricas usadas para evaluar la eficacia de los módelos son las versiones ponderadas (W) de precisión, recall y f1- score para la predicción exacta de la taxonomía, es decir, la predicción parcial de la taxonomía de un producto no cuenta como una predicción correcta. Además, se considera el f1- score ponderado como la métrica determinante para clasificar los modelos. Estas métricas son comúnmente usadas en problemas de clasificación jerárquica [11], [61], [62] ya que, reflejan mejor la calidad de clasificación en presencia de un dataset altamente desbalanceado [38]. Estas métricas se denotan de la siguiente manera:

$$\text{W-Precision} = \frac{1}{N}\sum_{i=1}^{K} n_i \cdot \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \qquad (1)$$

$$\text{W-Recall} = \frac{1}{N}\sum_{i=1}^{K} n_i \cdot \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \qquad (2)$$

$$\text{W-F1} = \frac{1}{N}\sum_{i=1}^{K} n_i \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \qquad (3)$$

Donde, se asume que existen K clases, ci |i = 1, 2, ..., K tanto en el dataset de entrenamiento como en el de pruebas, debido a la partición estratificada. El número de instancias verdaderas para cada clase se denota como ni (support), y el total de instancias es N = PK i=1 ni .

V. RESULTADOS

A. CLASIFICACIÓN PLANA

El enfoque de clasificación plana fue aplicado en los 7 algoritmos definidos en este estudio, los cuales fueron construidos según el proceso ilustrado en la Figura 5, y el resultado de cada uno de los experimentos ejecutados en la etapa de Pre-procesamiento y Feature Engineering se muestran en la Tabla 6.

TABLA 6: Resultados de la Preparación de Datos

Algorithm	Data Cleaning	Data Augmentation	Text Representation	Dimensionality Reduction
MNB	replace_symbols, remove_numeric_token, remove_stopw_smallw, custom_ngrams	100	CountVectorizer	NOT IMPROVED
MLR	replace_symbols, remove_stopw_smallw, stemming, custom_ngrams	100	CountVectorizer	NOT IMPROVED
LSVC	replace_symbols, remove_stopw_smallw, custom_ngrams	100	TF-IDF sublinear	NOT IMPROVED
RF	replace_symbols, replace_patterns, remove_numeric_token, remove_stopw_smallw, stemming	NOT IMPROVED	CountVectorizer	NOT IMPROVED
XGB	replace_symbols, remove_numeric_token, replace_patterns, lemmatization	NOT IMPROVED	CountVectorizer	1000
FT*	replace_symbols, remove_stopw_smallw	NOT IMPROVED	DOES NOT APPLY	DOES NOT APPLY
Ensemble (hard)*	replace_symbols, remove_stopw_smallw, custom_ngrams	100	TF-IDF sublinear	DOES NOT APPLY

** No cross-validation applied*

En esta tabla se observa que los algoritmos FastText y Voting Ensemble (hard) difieren del resto en la etapa de Feature Engineering, ambos recibieron la etiqueta 'NOT APPLY'. En el caso de FastText, se debe a que este usó su propio método de representación de texto y ajuste de hiperparámetros. Respecto al Ensemble, ninguno de los 3 algoritmos que lo componen (MNB, MLR, LSVC) mejoraron su rendimiento individual al reducir su dimensionalidad, tal y como se observa en las primeras filas de la misma tabla, por ello, tampoco se aplicó esta técnica en este algoritmo. Cabe mencionar que, tanto en la etapa de Preprocesamiento como de Feature Engineering se aplicó el método de cross-validation 5-fold usando solamente el dataset de entrenamiento. El dataset de pruebas fue usado recién en la evaluación final del modelo.

Asimismo, en base a los resultados de la Tabla 6, podemos afirmar que los métodos de Data Cleaning que más favorecen a los modelos de ML son replace_symbols, remove_stopw_smallw y custom_ngrams; además, los algoritmos de Random Forest y XGBoost muestran un mejor rendimiento cuando se agrega la técnica de stemming o lemmatization y, a su vez, se remueven los tokens numéricos; también, la técnica de stemming mejora el algoritmo MLR y la remoción de tokens numéricos mejora el algoritmo MNB. Del mismo modo, el método personalizado de ngrams es mejor que las técnicas de bi-gram o tri-gram en todos los algoritmos.

En lo que concierne al aumento de datos, los experimentos comprobaron que los algortimos alcanzaron su mejor rendimiento cuando θ era igual a 100, excepto para XGB, RF y FT, donde ningún valor de θ mejoró su desempeño. Por otro lado, el método de representación de texto que demostró mejores resultados en la mayoría de los algoritmos fue el CountVectorizer, excepto para el algoritmo LSVC y Voting Ensemble (hard), en los cuales el método de TFIDF sublineal supera a las otras 2 técnicas de vectorización. Por último, el método Truncated-SVD, usado para reducir la dimensionalidad, solamente mejoró el desempeño de XGB cuando se redujo a 1000 dimensiones.

Siguiendo el proceso descrito en la Figura 5, la última etapa corresponde a la selección de los modelos que implica el ajuste de los hiperparámetros y la evaluación de desempeño del algoritmo. El rango de valores usados para cada hiperparámetro se detalla en el Anexo así como los valores finales y los que fueron usados en el modelo final para cada algoritmo. Sumado a esto, es importante señalar que para el algoritmo FT y Ensemble (hard) la selección de los hiperparámetros es distinta, en caso del algoritmo FT se realizó usando su propio método de autoentrenamiento por 3 horas, y para el Ensemble (hard) se adoptaron los mismos hiperparámetros usados en los experimentos independientes de cada algoritmo.

TABLA 7: Rendimiento del modelo con enfoque plano (10-folds)

Algorithm	accuracy	waF1	waPrecision	waRecall
MNB	0.8097 ± 0.002	0.8045 ± 0.002	0.8072 ± 0.002	0.8097 ± 0.002
MLR	0.8151 ± 0.002	0.8132 ± 0.002	0.8154 ± 0.002	0.8151 ± 0.002
LSVC	0.8238 ± 0.002	0.8183 ± 0.002	0.8194 ± 0.002	0.8238 ± 0.002
RF	0.7895 ± 0.002	0.7818 ± 0.002	0.7846 ± 0.002	0.7895 ± 0.002
XGB	0.7717 ± 0.003	0.7643 ± 0.003	0.7625 ± 0.003	0.7717 ± 0.003
FT	0.7854 ± 0.003	0.7811 ± 0.003	0.7839 ± 0.002	0.7854 ± 0.003
Ensemble (hard)	**0.8254 ± 0.002**	**0.8197 ± 0.002**	**0.8210 ± 0.002**	**0.8254 ± 0.002**

En resumen, la Tabla 7 muestra los resultados de las métricas de evaluación para los 7 algoritmos con un enfoque de clasificación plano. Basados en estos resultados, notamos que el algoritmo con mejor rendimiento es el Ensemble (hard), seguido por LSVC y MLR, siendo los únicos algoritmos que sobrepasan el 80% en la métrica ponderada de F1-score.

B. CLASIFICACIÓN JERÁRQUICA

Como se evidencia en la Tabla 8, el modelo LSVC presenta el mejor rendimiento para el nivel 1, 2 y 3, mientras que en el nivel 4, el mejor modelo es el MLR superando por 0.62% al modelo LSVC. Basados en estos resultados, la arquitectura del modelo jerárquico (LCL) fue compuesta por 3 modelos LSVC y un modelo MLR, ilustrado también en la Figura 7.

TABLA 8: Rendimiento del modelo por nivel jerárquico (5-folds)

Level	Algorithm	accuracy	waF1	waPrecision	waRecall
	MNB	0,9282 ± 0.001	0,9284 ± 0.001	0,9288 ± 0.001	0,9282 ± 0.001
L1	MLR	0,9344 ± 0.000	0,9345 ± 0.000	0,9347 ± 0.000	0,9344 ± 0.000
	LSVC	0,9412 ± 0.001	**0,9411 ± 0.001**	0,9411 ± 0.001	0,9412 ± 0.001
	MNB	0,8877 ± 0.001	0,8877 ± 0.001	0,8900 ± 0.001	0,8877 ± 0.001
L2	MLR	0,8910 ± 0.002	0,8912 ± 0.002	0,8924 ± 0.002	0,8910 ± 0.002
	LSVC	0,9009 ± 0.002	**0,8999 ± 0.002**	0,8999 ± 0.002	0,9009 ± 0.002
	MNB	0,7947 ± 0.002	0,7754 ± 0.002	0,7663 ± 0.002	0,7947 ± 0.002
L3	MLR	0,8005 ± 0.001	0,7822 ± 0.001	0,7711 ± 0.001	0,8005 ± 0.001
	LSVC	0,8103 ± 0.001	**0,7883 ± 0.001**	0,7745 ± 0.001	0,8103 ± 0.001
	MNB	0,4093 ± 0.001	0,2978 ± 0.001	0,2446 ± 0.001	0,4093 ± 0.001
L4	**MLR**	0,4157 ± 0.001	**0,3102 ± 0.001**	0,2627 ± 0.001	0,4157 ± 0.001
	LSVC	0,4190 ± 0.001	0,3040 ± 0.001	0,2500 ± 0.001	0,4190 ± 0.001

Asimismo, se revela que la eficacia de los modelos decrece conforme el nivel de la jerarquía aumenta, sin embargo, existe un descenso abrupto en el nivel 4, debido principalmente a que las clases son más específicas en este nivel, por ende, es más difícil encontrar patrones que permitan distinguir entre las mismas, además, también se debe al aumento de número de clases (2) y la escasa representación por clase. Este comportamiento es usual en los modelos jerárquicos [12], [25].

Los resultados propios del modelo jerárquico (LCL) se exponen en las Tablas 9 y 10, la diferencia entre ambas tablas es cómo usan los datos de entrenamiento, la Tabla 9 usa el enfoque hier-approach-1, mientras que la otra tabla usa el enfoque hier-approach-2, de este modo, usando hier-approach-1 obtenemos un f1-score ponderado igual a 39.26%, mientras que usando hier-approach-2 la versión ponderada de f1-score alcanza un 77.23%. Comprando así que la forma forma de usar los datos de entrenamiento es el hier-approach-2, el cual usa todos los productos del dataset en cada modelo y asigna la etiqueta ’0’ a los productos que no pertenecen a una categoría en el nivel de la jerarquía que se predice.

TABLA 9: Rendimiento del modelo Plano-Jerárquico usando hier-approach-1 (10-folds).

Model	accuracy	waF1	waPrecision	waRecall
lsvc_cat1	0,9418 ± 0.001	0,9418 ± 0.001	0,9418 ± 0.001	0,9418 ± 0.001
lsvc_cat2	0,9012 ± 0.002	0,9002 ± 0.002	0,9002 ± 0.002	0,9012 ± 0.002
lsvc_cat3	0,8112 ± 0.001	0,7889 ± 0.002	0,7751 ± 0.001	0,8112 ± 0.001
mlr_cat4	0,4118 ± 0.001	0,3090 ± 0.001	0,2623 ± 0.001	0,4118 ± 0.001
hierarchical	0,3810 ± 0.001	0,3926 ± 0.001	0,4100 ± 0.001	0,3810 ± 0.001
flat	**0,8252 ± 0.002**	**0,8196 ± 0.002**	0,8209 ± 0.002	**0,8252 ± 0.002**
flat_hierarchical	0,8239 ± 0.002	0,8190 ± 0.002	**0,8210 ± 0.002**	0,8239 ± 0.002

TABLA 10: Rendimiento del modelo Plano-Jerárquico usando hier-approach-2 (10-folds).

Model	accuracy	waF1	waPrecision	waRecall
lsvc_cat1	0,9418 ± 0.001	0,9418 ± 0.001	0,9418 ± 0.001	0,9418 ± 0.001
lsvc_cat2	0,9012 ± 0.002	0,9002 ± 0.002	0,9002 ± 0.002	0,9012 ± 0.002
lsvc_cat3	0,8521 ± 0.001	0,8482 ± 0.002	0,8486 ± 0.001	0,8521 ± 0.001
mlr_cat4	0,7951 ± 0.002	0,8080 ± 0.002	0,8452 ± 0.002	0,7951 ± 0.002
hierarchical	0,7283 ± 0.002	0,7723 ± 0.002	**0,8606 ± 0.001**	0,7283 ± 0.002
flat	0,8252 ± 0.002	0,8196 ± 0.002	0,8209 ± 0.002	0,8252 ± 0.002
flat_hierarchical	**0,8263 ± 0.002**	**0,8211 ± 0.002**	0,8225 ± 0.002	**0,8263 ± 0.002**

C. CLASIFICACIÓN PLANO-JERÁRQUICA

Por otro lado, en las Tablas 9 y 10 también se evidencia que al usar el algoritmo 2 para unir las predicciones del modelo plano y el modelo jerárquico (LCL), los resultados divergen según el uso de los datos de entrenamiento; es decir, cuando usamos el hier-approach-1 el modelo flat_hierarchical alcanza un f1-score ponderado igual a 81.90%, cuyo rendimiento es menor que el modelo con enfoque plano que alcanza un 81.96%. Por el contrario, cuando se utiliza el hier-approach-2, el modelo propuesto flat_hierarchical logra un f1-score ponderado de 82.11%, el cual supera por 0.15% al modelo flat y por 4.88% al modelo hierarchical.

VI. DISCUSIÓN

De acuerdo a la revisión literaria conducida por el autor, los resultados de la presente investigación concuerdan con estudios previos en esta área. En primer lugar, se demostró que un modelo con enfoque plano presenta mejores resultados que un modelo con enfoque jerárquico (LCL), lo cual también fue comprobado por de Ding et al. [36] quien usó algoritmos tradicionales de ML y un clasificador jerárquico de tipo LCL, y por Krishnan y Amarthaluri [5] usando algoritmos de Deep Learning y un clasificador jerárquico de tipo LCN. No obstante, otros autores demostraron que un modelo con enfoque jerárquico basado en una arquitectura de redes neuronales puede presentar un mejor rendimiento, lo cual no fue contemplado en este estudio [13], [21], así también, Ozyegen et al. [12] sugiere el uso de modelos pre entrenados y transformers para aumentar la eficacia del modelo jerárquico (LCL).

En segundo lugar, cuando comparamos el desempeño de los algoritmos tradicionales de ML, LinearSVC sobresale entre todos, estos resultados coinciden con los obtenidos por Allweyer et al. [25] y Goumy y Mejri [63], el primero, trabajó con un dataset similar en tamaño y profundidad de la estructura jerárquica, y desarrolló un modelo jerárquico (LCL), mientras que, el segundo, trabajó con un dataset mucho más grande, desarrolló un modelo con enfoque plano y otro modelo jerárquico que combina LCL y LCPN; ambos estudios concluyeron que el algoritmo LinearSVC demuestró el mejor desempeño.

En tercer lugar, hasta donde se sabe, no se han encontrado artículos que investiguen la combinación del enfoque plano y jerárquico en un modelo de ML, lo cual brinda mayor importancia a esta investigación. No obstante, el presente estudio tiene algunas limitaciones, tales como la experimentación con modelos pre entrenados para la representación de texto

(BERT, RoBERTa, ELMO) [53], modelos de clasificación basados en Deep Learning o Large Language Models (LLM), además, en este estudio no se utilizó el dataset original, compuesto por más de 1 millón de productos, y útil para modelos más complejos.

Por último, el dataset que disponibiliza este estudio es el primero en la comunidad científica con sus características: datos en español, de acceso público, de gran escala y con una estructura jerárquica de 6 niveles. Entre los otros datasets de acceso público y usados para la clasificación jerárquica de productos, podemos mencionar el dataset de gran escala proporcionado por SIGIR 2018 eCom Rakuten Data Challenge, compuesto por 1 millón de productos en inglés [61], igualmente, en el MWPD Challenge se facilitó un dataset jerárquico de 13K productos [62] y Brinkmann y Bizer [13] usaron el dataset ICECAT/WDC2225 , que posee más de 750K productos. No obstante, cabe notar que, los datasets mencionados e incluyendo el que se presenta en esta investigación, son de tipo single-model, es decir no se adaptan para modelos multimodales (texto e imagen).

Aunque este estudio ha arrojado luz sobre ciertos aspectos, hay otros aún sin explorar que merecen una investigación más detallada. En este sentido, investigaciones futuras pueden contemplar el uso de transformers y LLM en los modelos de clasificación y comprobar si la combinación de ambos enfoques aún puede mejorar el rendimiento individual. Además, también es necesario investigar el rendimiento de las otras variaciones del clasificador local como LCPN y LCPN. Otra posible investigación podría basarse en modelos multilinguísticos o que usen métodos de Transfer Learning, ya que, es posible recolectar datos en portugués de la misma plataforma de comercio electrónico. Finalmente, considerando los diversos caminos de investigación a futuro, se cree que el dataset propuesto motivará y facilitará tanto la investigación multilinguística como la investigación de modelos más complejos que necesitan de datasets de gran escala.

VII. ANÁLISIS DE ERROR

En esta sección se realiza un análisis detallado de los errores derivados de nuestro modelo de clasificación jerárquica. El objetivo es entender las áreas en las que el modelo puede estar fallando y proporcionar valiosas recomendaciones para futuras investigaciones.

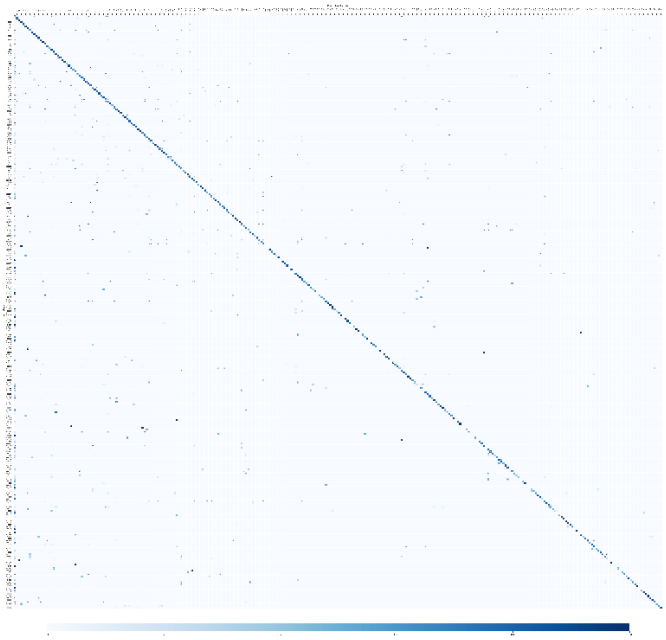


FIGURA 8: Matriz de confusión para los nodos finales

La Figura 8 muestra la matriz de confusión del modelo propuesto en esta investigación, considerando las 543 clases finales del dataset D_s. A partir de esta figura notamos que en la mayoría de clases donde nuestro modelo es ineficaz se debe a que existe una estrecha similitud con alguna otra clase. Por ejemplo, solo el 4% de instancias de la clase 1648_430687_445198_445199 que hace referencia a Computación > Laptops y Accesorios > Repuestos para Laptops > Memorias RAM para Laptops fueron predichas correctamente y el 93% de ellas fueron asignadas a la clase final 1648_444889_1694 que corresponde a Computación > Componentes de PC > Memorias RAM. Por lo tanto, esta estrecha similitud entre ambas clases finales es una de las razones por las que el modelo no logra ser más eficaz.

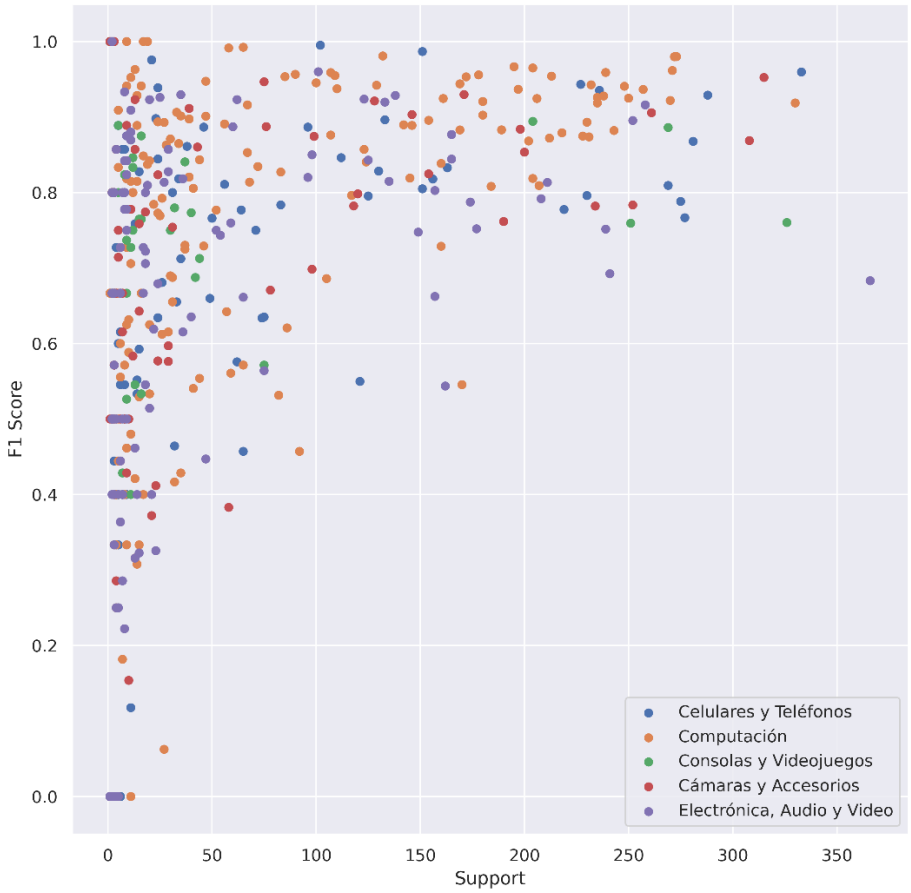


FIGURA 9: F1-score por taxonomía

Por otro lado, en base a la Figura 9 existe una clara evidencia que cuando una clase posee más instancias el modelo puede predecir mejor la taxonomía de nuevos productos. Aunque algunas clases con escasa representación obtuvieron un rendimiento aceptable, el 14% de las clases finales (78) alcanzaron un f1-score igual a 0, cuyo número de instancias en el dataset de pruebas varía entre 1 y 11, reforzando así la importancia de la cantidad de representación por clase.

VIII. CONCLUSIONES

Tras un análisis exhaustivo de los datos y las evidencias, se pueden extraer varias conclusiones significativas. En primer lugar, se comprueba que el mejor modelo de ML no ensemblado es el LinearSVC combinado con el método TF-IDF sublineal y que los métodos de

limpieza más efectivos para este dataset son: replace_symbols, remove_stopw_smallw y custom_ngrams, resaltando que el método custom_ngrams proporciona mejores características que los comunes: uni-gram y bi-gram.

En segundo lugar, se comprueba que el modelo con enfoque plano, Voting Ensemble (hard), es mejor que el modelo con enfoque jerárcuico (LCL), superándolo por 5.39% según la métrica ponderada de f1-score. Así también, se concluyó que la mejor estrategia en la forma de uso de los datos de entrenamiento en el modelo jerárquico (LCL) es el hierapproach-2, la cual desta sobre el hier-approach-1 por un 33.79%.

Por último, el modelo de ML sugerido en este estudio, el cual combina el enfoque plano y jerárquico para la predicción de la taxonomía de un producto, supera por 0.15% al modelo flat_ensemble y por 4.88% al modelo hierarchical. Demostrando así que es una potencial solución para mejorar un sistema de clasificación jerárquico.

REFERENCIAS

[1] V. Gupta, H. Karnick, A. Bansal, and P. Jhala, “Product classification in e-commerce using distributional semantics,” pp. 536–546, The COLING 2016 Organizing Committee, 12 2016.

[2] P. Das, Y. Xia, A. Levine, G. D. Fabbrizio, and A. Datta, “Large-scale taxonomy categorization for noisy product listings,” pp. 3885–3894, 2016.

[3] V. Umaashankar, G. S. S, and A. Prakash, “Atlas: A dataset and benchmark for e-commerce clothing product categorization,” CoRR, vol. abs/1908.08984, 2019.

[4] Z. Kozareva, “Everyone likes shopping! multi-class product categorization for e-commerce,” pp. 1329–1333, Association for Computational Linguistics, 5 2015.

[5] A. Krishnan and A. Amarthaluri, “Large scale product categorization using structured and unstructured attributes,” CoRR, vol. abs/1903.04254, 3 2019.

[6] R. M. Pereira, Y. M. Costa, and C. N. Silla, “Handling imbalance in hierarchical classification problems using local classifiers approaches,” Data Mining and Knowledge Discovery, vol. 35, pp. 1564–1621, 7 2021.

[7] E. Lehmann, A. Simonyi, L. Henkel, and J. Franke, “Bilingual transfer learning for online product classification,” pp. 21–31, Association for Computational Linguistics, 12 2020.

[8] W. Zhang, Y. Lu, B. Dubrov, Z. Xu, S. Shang, and E. Maldonado, “Deep hierarchical product classification based on pre-trained multilingual knowledge,” IEEE - The Bulletin of the Technical Committee on Data Engineering, 2021.

[9] J. W. Ha, H. Pyo, and J. Kim, “Large-scale item categorization in ecommerce using multiple recurrent neural networks,” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 13-17-August-2016, pp. 107–115, 8 2016.

[10] I. Hasson, S. Novgorodov, G. Fuchs, and Y. Acriche, “Category recognition in e-commerce using sequence-to-sequence hierarchical classification,” WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 902–905, 8 2021.

[11] F. Liu, D. Chen, X. Du, R. Gao, and F. Xu, “Mep-3m: A large-scale multi-modal e-commerce product dataset,” Pattern Recognition, vol. 140, p. 109519, 8 2023.

[12] O. Ozyegen, H. Jahanshahi, M. Cevik, B. Bulut, D. Yigit, F. F. Gonen, and A. Ba¸sar, “Classifying multi-level product categories using dynamic masking and transformer models,” Journal of Data, Information and Management 2022 4:1, vol. 4, pp. 71–85, 4 2022.

[13] A. Brinkmann and C. Bizer, “Improving hierarchical product classification using domain-specific language modelling,” IEEE Data Eng. Bull., vol. 44, pp. 14–25, 2021.

[14] L. Yang, E. Shijia, S. Xu, and Y. Xiang, “Bert with dynamic masked softmax and pseudo labeling for hierarchical product classification,” 2020.

[15] T. M. Tashu, S. Fattouh, P. Kiss, and T. Horváth, “Multimodal e-commerce product classification using hierarchical fusion,” pp. 279–284, 2022.

[16] Q. Chen, Z. Shi, Z. Zuo, J. Fu, and Y. Sun, “Two-stream hybrid attention network for multimodal classification,” Proceedings - International Conference on Image Processing, ICIP, vol. 2021-September, pp. 359–363, 2021.
[17] Y. Bi, S. Wang, and Z. Fan, “A multimodal late fusion model for ecommerce product classification,” 8 2020.
[18] L. Tan, M. Y. Li, and S. Kok, “E-commerce product categorization via machine translation,” ACM Trans. Manage. Inf. Syst., vol. 11, 7 2020.
[19] C. N. Silla and A. A. Freitas, “A survey of hierarchical classification across different application domains,” Data Mining and Knowledge Discovery, vol. 22, pp. 31–72, 1 2011.
[20] H. B. Borges, C. N. Silla, and J. C. Nievola, “An evaluation of globalmodel hierarchical classification algorithms for hierarchical classification problems with single path of labels,” Computers & Mathematics with Applications, vol. 66, pp. 1991–2002, 12 2013.
[21] D. Gao, W. Yang, H. Zhou, Y. Wei, Y. Hu, and H. Wang, “Deep hierarchical classification for category prediction in e-commerce system,” CoRR, vol. abs/2005.06692, 5 2020.
[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of Tricks for Efficient Text Classification,” 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference, vol. 2, pp. 427–431, jul 2016.
[23] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, “A survey on text classification algorithms: From text to predictions,” Information 2022, Vol. 13, Page 83, vol. 13, p. 83, 2 2022.
[24] M. Harth, C. Schorr, and R. Krieger, “A hierarchical multi-level product classification workbench for retail,” vol. 2738, pp. 59–69, CEUR-WS.org, 2020.
[25] O. Allweyer., C. Schorr., R. Krieger., and A. Mohr., “Classification of products in retail using partially abbreviated product names only,” pp. 67– 77, SciTePress, 2020.
[26] L. Akritidis, A. Fevgas, and P. Bozanis, “Effective products categorization with importance scores and morphological analysis of the titles,” pp. 213– 220, 11 2018.
[27] J. Dai, T. Wang, and S. Wang, “A deep forest method for classifying ecommerce products by using title information,” pp. 1–5, 2 2020.
[28] S. A. Oyewole and O. O. Olugbara, “Product image classification using eigen colour feature with ensemble machine learning,” Egyptian Informatics Journal, vol. 19, pp. 83–100, 2018.
[29] Y. Tang, F. Borisyuk, S. Malreddy, Y. Li, Y. Liu, and S. Kirshner, “Msuru: Large scale e-commerce image classification with weakly supervised search data,” p. 2518–2526, Association for Computing Machinery, 2019.
[30] Y. Seo and K. shik Shin, “Hierarchical convolutional neural networks for fashion image classification,” Expert Systems with Applications, vol. 116, pp. 328–339, 2019.
[31] E. Verma, S. Chakraborty, and V. Motupalli, “Deep multi-level boosted fusion learning framework for multi-modal product classification,” 2018.
[32] C. Chavaltada, K. Pasupa, and D. R. Hardoon, “Combining multiple features for product categorisation by multiple kernel learning,” pp. 3–12, Springer International Publishing, 2019.
[33] M. M. Hafez, A. F. Vilas, R. P. D. Redondo, and H. O. Pazó, “Classification of retail products: From probabilistic ranking to neural networks,” Applied Sciences, vol. 11, 2021.
[34] M. Skinner, “Product categorization with lstms and balanced pooling views,” 2018.
[35] D. Vandic, F. Frasincar, and U. Kaymak, “A framework for product description classification in e-commerce,” J. Web Eng., vol. 17, pp. 1–27, 3 2018.
[36] Y. Ding, M. Korotkiy, B. Omelayenko, V. Kartseva, V. Zykov, M. Klein, E. Schulten, and D. Fensel, “Goldenbullet: Automated classification of product data in e-commerce,” in Proceedings of the 5th international conference on business information systems, vol. 5, 2002.
[37] C. Chavaltada, K. Pasupa, and D. R. Hardoon, “A comparative study of machine learning techniques for automatic product categorisation,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10261 LNCS, pp. 10–17, 2017.

[38] B. Oancea, “Automatic product classification using supervised machine learning algorithms in price statistics,” Mathematics 2023, Vol. 11, Page 1588, vol. 11, p. 1588, 3 2023.
[39] C. Sun, N. Rampalli, F. Yang, and A. Doan, “Chimera: Large-scale classification using machine learning, rules, and crowdsourcing,” Proc. VLDB Endow., vol. 7, pp. 1529–1540, 8 2014.
[40] Y. Xia, A. Levine, P. Das, G. D. Fabbrizio, K. Shinzato, and A. Datta, “Large-scale categorization of japanese product titles using neural attention models,” pp. 663–668, Association for Computational Linguistics, 4 2017.
[41] H. Chen, J. Zhao, and D. Yin, “Fine-grained product categorization in e-commerce,” International Conference on Information and Knowledge Management, Proceedings, pp. 2349–2352, 11 2019.
[42] S. Suzuki, Y. Iseki, H. Shiino, H. Zhang, A. Iwamoto, and F. Takahashi, “Convolutional neural network and bidirectional lstm based taxonomy classification using external dataset at sigir ecom data challenge,” vol. 2319, 2018.
[43] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2016.
[44] S. Xu, Y. Li, and Z. Wang, “Bayesian multinomial naïve bayes classifier to text classification,” Lecture Notes in Electrical Engineering, vol. 448, pp. 347–352, 2017.
[45] A. Genkin, D. D. Lewis, and D. Madigan, “Large-scale bayesian logistic regression for text categorization,” Technometrics, vol. 49, no. 3, pp. 291– 304, 2007.
[46] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, “Sparse multinomial logistic regression: fast algorithms and generalization bounds,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp. 957–968, 2005.
[47] V. Vapnik, “An overview of statistical learning theory,” IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 988–999, 1999.
[48] C. N. Kamath, S. S. Bukhari, and A. Dengel, “Comparative study between traditional machine learning and deep learning approaches for text classification,” Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018, 8 2018.
[49] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” the Journal of machine Learning research, vol. 9, pp. 1871–1874, 2008.
[50] C. Heistracher, F. Mignet, and S. Schlarb, “Machine learning techniques for the classification of product descriptions from darknet marketplaces.,” pp. 128–137, 2020.
[51] T. K. Ho, “Random decision forests,” in Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282 vol.1, 1995.
[52] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 10 2001.
[53] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models,” Transactions on Asian and LowResource Language Information Processing, vol. 20, 6 2021.
[54] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.
[55] Abdullah-All-Tanvir, I. A. Khandokar, A. K. M. Islam, S. Islam, and S. Shatabda, “A gradient boosting classifier for purchase intention prediction of online shoppers,” Heliyon, vol. 9, p. 15163, 4 2023.
[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pas sos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikitlearn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[57] O. Sagi and L. Rokach, “Ensemble learning: a survey,” WIREs Data Mining and Knowledge Discovery, vol. 8, 2018.
[58] C. A. Gonçalves, A. S. Vieira, C. T. Gonçalves, R. Camacho, E. L. Iglesias, and L. B. Diz, “A novel multi-view ensemble learning architecture to improve the structured text classification,” Information 2022, Vol. 13, Page 283, vol. 13, p. 283, 6 2022.
[59] R. Wirth and J. Hipp, “Crisp-dm: Towards a standard process model for data mining,” in Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, vol. 1, pp. 29–39, Manchester, 2000.
[60] A. Cevahir and K. Murakami, “Large-scale multi-class and hierarchical product categorization for an e-commerce giant,” pp. 525–535, The COLING 2016 Organizing Committee, 12 2016.
[61] Y.-C. Lin, P. Das, and A. Datta, “Overview of the sigir 2018 ecom rakuten data challenge,” vol. 2319, CEUR-WS.org, 2018.
[62] Z. Zhang, C. Bizer, R. Peeters, and A. Primpeli, “Mwpd2020: Semantic web challenge on mining the web of html-embedded product data,” vol. 2720, CEUR-WS.org, 2020.
[63] S. Goumy and M.-A. Mejri, “Ecommerce product title classification,” vol. 2319, CEUR-WS.org, 2018

HAROLD COTACALLAPA received the B.S. degree in Systems Engineering from Universidad Peruana Unión, Perú. He has experience working as a Software Developer and Data Analyst. His main research interests include Machine Learning applications in neuroscience, climate change and finances.

NEMIAS SABOYA (Member, IEEE) holds a B.S. degree in Systems Engineering from Universidad Peruana Unión (UPeU, Peru) and a M.S. degree in Computer and Systems Engineering, specialization: Information Technology Management (USMP, Peru). He is currently a Ph.D. candidate in Systems Engineering at Universidad Peruana Unión (UPeU, Peru). His main research interests include data governance, statistical Machine Learning, IT and data science.

PAULO CANAS RODRIGUES is a Professor of Statistics at the Federal University of Bahia and the Head and Principal Investigator of the Statistical Learning Laboratory (SaLLy). He completed his Ph.D. in Statistics at the Nova University of Lisbon, Portugal (2012), and his Aggregation (Habilitation) in Mathematics, with a specialization in Statistics and Stochastic Processes, at the Lisbon University, Portugal (2019). His research interests include statistical learning, time series forecasting, and data science in general.

RODRIGO SALAS F. (Senior Member, IEEE) received the B.S. and MSc. degrees in Informatics Engineering and the Dr. Eng. degree in informatics from the Federico Santa María Technical University (UTFSM) in Chile, in 2001, 2002 and 2010, respectively. From 2002 to 2004, he was a research assistant with the Informatics Department, UTFSM. Since 2004, has been with the Universidad de Valparaíso, where he is currently a Full Professor of the Biomedical Engineering

School and teaches in Data Mining, Probability and Statistics, and Machine Learning. Dr. Salas is main researcher at Millennium Institute for Intelligent Healthcare Engineering (i-HEALTH), and main researcher at the Center of Research and Development in Health Engineering (CINGS-UV). His research interests include Artificial Intelligence, Data Science, Computational Statistics, Decision Support Systems, Intelligent Systems and their applications to finance, air pollution, healthcare and medicine.

JAVIER LINKOLK LÓPEZ GONZALES (Member, IEEE) received the B.S. degree in Statistical and Informatics Engineering from Universidad Peruana Unión (UPeU, Perú) and the M.Sc. degree in Metrology from Pontifical Catholic University of Rio de Janeiro (PUC-Rio, Brazil). Likewise, received the Ph.D. degree in Statistics of Universidad de Valparaíso (UV, Chile). His main research interests include pattern recognition in Machine Learning, air pollution with Deep Learning techniques, and Time Series with Singular Spectrum Analysis. He is an associate professor of Facultad de Ingeniería y Arquitectura, Universidad Peruana Unión.

APÉNDICE. A. AJUSTE DE HIPERPARÁMETROS

TABLA A1: Ajuste de hiperparámetros para los modelos de machine learning

Algorithm	Search space (cv=5)	Best hyperparameters	Model
MNB	min_df: [1, 2, 3, 4, 6, 10], alpha: [0.001, 0.01, 0.05, 0.1, 1.0, 5.0, 10.0]	min_df'=1 alpha=0.01	CountVectorizer() MultinomialNB(alpha=0.01)
MLR	min_df: [1,2,3,4,8], max_features: [1000, 10000, 50000, 100000, 150000, None] C: [0.5, 1.0, 2.0, 3.0, 5.0, 10.0], solver: ['saga','sag'], class_weight: ['balanced',None]	min_df': 1 max_features: None C: 5.0, solver: 'sag' class_weight': 'balanced'	CountVectorizer() LogisticRegression(C=5.0, solver='sag' class_weight='balanced', max_iter=1000, multi_class='multinomial', n_jobs=-1, random_state=42)
LSVC	min_df: [1,2,3,4,6,10], max_features: [1000, 10000, 50000, 100000, None] C: [1.0, 2.0, 3.0, 5.0, 10.0, 15.0], loss: ['hinge', 'squared_hinge'], class_weight: ['balanced', None], dual: [True, False], max_iter: [100,1000,10000,100000]	max_features: None, min_df: 1, C: 2.0, loss:'hinge', max_iter:100, class_weight: None, dual: True	CountVectorizer() TfidfTransformer(sublinear_tf=True) LinearSVC(C=2.0, loss='hinge', max_iter=100, random_state=42, verbose=3)
RF	min_df: [1,3,4,8,10], max_features: [1000, 10000, 100000, None] criterion: ['gini','entropy','log_loss'], max_depth: [50,100,150,300], n_estimators: [70,100,200]	max_features: None, min_df: 8, criterion: 'gini', max_depth: 300, n_estimators: 70	CountVectorizer(min_df=8) RandomForestClassifier(max_depth=300, n_estimators=70, n_jobs=-1, random_state=42)
XGB	min_df: [1,3,4,8,10], max_features: [2000, 5000, 10000, 100000, None] max_depth: [1,2,3,4,5,7,10,12] objective: ['multi:softprob','multi:softmax'], min_child_weight: [1,2,3,5,7,10], learning_rate:[0.05,0.07,0.1,0.2,0.3,0.4,0.5] n_estimators: [70,100,200]	max_features: 10000, min_df: 3, max_depth: 7, objective: 'multi:softmax', min_child_weight: 10 learning_rate:0.2 n_estimators:500	CountVectorizer(max_features=10000, min_df=3) TruncatedSVD(n_components=1000) MinMaxScaler(), XGBClassifier(learning_rate=0.2, max_depth=7, min_child_weight=10, n_estimators=500, num_class=475, objective='multi:softmax', n_jobs=-1))
FT	NOT APPLY	dim=200, epochs=100, lr=0.05	dim=200, epochs=100, lr=0.05
Ensemble (hard)	NOT APPLY	NOT APPLY	VotingClassifier(estimators=[('svm', svm_model), ('nb', nb_model), ('lr', lr_model)], voting='hard')

EVIDENCIA DE LA SUMISIÓN DEL ARTÍCULO CIENTÍFICO

IEEE Access - Manuscript ID Access-2023-33315

IA

IEEE Access <onbehalfof@manuscriptcentral.com>

To: Nemias Saboya; Harold Cotacallapa; paulocanas@gmail.com; rodrigo.salas@uv.cl; javier.lopez@postgrado.uv.cl

Wed 10/18/2023 6:46 PM

18-Oct-2023

Dear Mr. SABOYA:

Your manuscript entitled "A flat-hierarchical approach based on machine learning model for e-commerce product classification" has been successfully submitted online and is presently being given full consideration for publication in IEEE *Access*.

As a reminder, IEEE *Access* is a fully open access journal. Open Access provides unrestricted access to published articles via IEEE *Xplore*. In lieu of paid subscriptions, authors are required to pay an article processing charge of $1,950 (plus applicable local taxes) after the article has been accepted for publication.

Your manuscript ID is Access-2023-33315. Please mention the manuscript ID in all future correspondence to the IEEE *Access* Editorial Office. The submitting author can view the manuscript status at any time by checking their author dashboard on the IEEE Author Portal. If the submitting author needs to update their email address after submission, please reach out to ieeeaccesseic@ieee.org so we can assist you in doing so.

Thank you again for submitting your manuscript to IEEE *Access*.

Sincerely,

IEEE *Access* Editorial Office

Reply | Reply all | Forward

Manuscripts I Have Co-Authored

STATUS	ID	TITLE	CREATED	SUBMITTED
Contact Journal EIC: Abbott, Derek ADM: Not Assigned • Under Review	Access-2023-33315 (REX-PROD-2-ED690274-4242-43A7-8CB8-697A615F4BC5-26E078BD-436F-45A5-98FD-9C9AFA9FF70A-16773)	A flat-hierarchical approach based on machine learning model for e-commerce product classification View Submission Submitting Author: SABOYA, NEMIAS	18-Oct-2023	18-Oct-2023

"AÑO DEL BICENTENARIO DEL PERÚ: 200 AÑOS DE INDEPENDENCIA"

RESOLUCIÓN N° 0746/A-2021/UPeU-FIA-CF-T

Lima, Ñaña 27 de octubre de 2021

VISTO:

El expediente de **Harold Enrique Cotacallapa Mamani**, identificado(a) con Código Universitario Nº 201612707, de la Escuela Profesional de Ingeniería de Sistemas de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión;

CONSIDERANDO:

Que la Universidad Peruana Unión tiene autonomía académica, administrativa y normativa, dentro del ámbito establecido por la Ley Universitaria Nº 30220 y el Estatuto de la Universidad;

Que la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, mediante sus reglamentos académicos y administrativos, ha establecido las formas y procedimientos para la aprobación e inscripción del perfil de proyecto de tesis en formato artículo y la designación o nombramiento del asesor para la obtención del título profesional;

Que **Harold Enrique Cotacallapa Mamani**

, ha solicitado: la inscripción del perfil de proyecto de tesis titulado "Modelo de aprendizaje supervisado para la clasificación jerárquica de productos tecnológicos en e-commerce" y la designación del Asesor, encargado de orientar y asesorar la ejecución del perfil de proyecto de tesis en formato artículo;

Estando a lo acordado en la sesión del Consejo de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, celebrada el 27 de octubre de 2021, y en aplicación del Estatuto y el Reglamento General de Investigación de la Universidad;

SE RESUELVE:

Aprobar el perfil de proyecto de tesis en formato artículo titulado **"Modelo de aprendizaje supervisado para la clasificación jerárquica de productos tecnológicos en e-commerce"** y disponer su inscripción en el registro correspondiente, designar al **Mg. Nemias Saboya Rios** como ASESOR para que oriente y asesore la ejecución del perfil de proyecto de tesis en formato artículo el cual fue dictaminado por: **Mg. Fredy Abel Huanca Torres y Mg. Danny Lévano Rodríguez**, otorgándoles un plazo máximo de doce (12) meses para la ejecución.

Regístrese, comuníquese y archívese.

Dra. María Vallejos Atalaya de Cornejo
DECANA

Dra. Erika Inés Acuña Salinas
SECRETARIA ACADÉMICA

cc:
-Interesado
Asesor
Dirección General de Investigación
Archivo