UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

TEMA:

Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning.

AUTORES:

Cañar Benavides, José Alejandro

Iglesias Muñoz, Valeria Abigail

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE LICENCIADO EN NEGOCIOS INTERNACIONALES

TUTOR:

Ing. Carrera Buri, Félix Miguel, Mgs

Guayaquil, Ecuador

22 de agosto del 2026

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

CERTIFICACIÓN

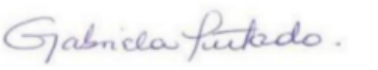
Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Cañar Benavides, José Alejandro e Iglesias Muñoz, Valeria Abigail**, como requerimiento para la obtención del título de **Licenciados en Negocios Internacionales**.

TUTOR (A)

f. _____

Ing. Carrera Buri, Félix Miguel, Mgs

DIRECTOR DE LA CARRERA



f. _____

Ing. Hurtado Cevallos, Gabriela Elizabeth, Mgs

Guayaquil, a los 22 días del mes de agosto del año 2026.

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **Cañar Benavides, José Alejandro e Iglesias Muñoz, Valeria Abigail**

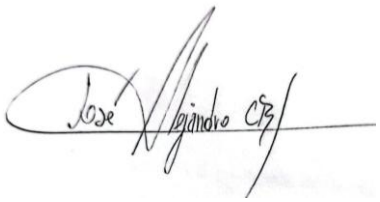
DECLARAMOS QUE:

El Trabajo de Titulación, **Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning,** previo a la obtención del título de **Licenciados en Negocios Internacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 22 días del mes de agosto del año 2026.

LOS AUTORES

f. 

Cañar Benavides, José Alejandro

f. _____

Iglesias Muñoz, Valeria Abigail

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

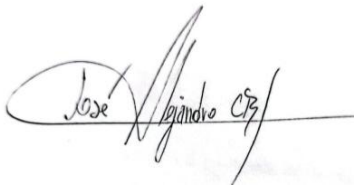
AUTORIZACIÓN

Nosotros, **Cañar Benavides, José Alejandro e Iglesias Muñoz, Valeria Abigail**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 22 días del mes de agosto del año 2026.

LOS AUTORES



f. _____

Cañar Benavides, José Alejandro

f. _____

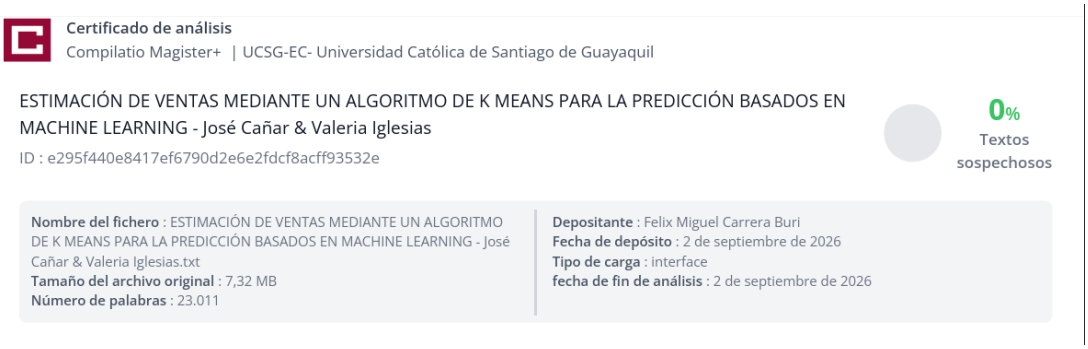
Iglesias Muñoz, Valeria Abigail

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

REPORTE COMPILATIO



Certificado de análisis
Compilatio Magister+ | UCSG-EC- Universidad Católica de Santiago de Guayaquil

ESTIMACIÓN DE VENTAS MEDIANTE UN ALGORITMO DE K MEANS PARA LA PREDICCIÓN BASADOS EN MACHINE LEARNING - José Cañar & Valeria Iglesias

ID : e295f440e8417ef6790d2e6e2fdcf8acff93532e

0% Textos sospechosos

Nombre del fichero : ESTIMACIÓN DE VENTAS MEDIANTE UN ALGORITMO DE K MEANS PARA LA PREDICCIÓN BASADOS EN MACHINE LEARNING - José Cañar & Valeria Iglesias.txt
Tamaño del archivo original : 7,32 MB
Número de palabras : 23.011

Depositante : Felix Miguel Carrera Buri
Fecha de depósito : 2 de septiembre de 2026
Tipo de carga : interface
fecha de fin de análisis : 2 de septiembre de 2026

f. _____

Ing. Carrera Buri, Félix Miguel, Mgs

TUTOR

AGRADECIMIENTO

JOSÉ ALEJANDRO CAÑAR BENAVIDES

En primer lugar, quiero agradecer a Dios por darme la oportunidad de estudiar, por brindarme los recursos necesarios para superar cada etapa y proceso en mi vida, por ser mi refugio, mi amparo, mi padre y mi amigo.

A mi mami Silvia, por ser la mejor madre que pude haber tenido en el mundo. Gracias por tu ejemplo de honestidad y esfuerzo, por brindarme las fuerzas necesarias y creer en mi cuando incluso yo no he creído. Gracias por llevarme a ver más allá de lo que mis ojos pueden ver.

A mi abuelita, mi mami Edith, gracias por amarme como a un hijo, "mi último" como ella suele decirme. Gracias por cubrir cada uno de mis pasos con sus oraciones y por ser reflejo del amor de cristo en mi vida. Gracias por sus consejos, por nunca dejar de bendecirme y amarme incondicionalmente.

A mis tíos maternos, gracias a cada uno de ustedes por ser familia y por el ejemplo de amor, solidaridad y unión que practican a diario. Cada uno desde su propia perspectiva con cada consejos y conversaciones me ha llevado a mejorar y creer en mi. Los amo y agradezco todo lo que han aportado a mi vida.

A mi amiga y compañera, no solo de tesis, sino de carrera Vale. Gracias por acompañarme durante esta hermosa etapa universitaria y brindarme tu amistad genuina y sincera. Gracias por cada conversación, cada consejo, cada explicación académica, cada competencia y cada experiencia compartida durante estos años. Gracias por convertirte en mi hermana de vida.

A mis amigos de la carrera, Johnny, Ronnie, Zach, Ash, Belen por acompañarme bridarme su amistad y formar parte de esta hermosa etapa. Gracias por todas las risas y llantos que hicieron mi vida universitaria un recuerdo que siempre llevare conmigo.

A mi tutor, Félix Carrera, por ser ejemplo de profesionalismo, por aconsejarme y prepararme con una visión hacia los desafíos reales que me esperan en el ámbito profesional. Gracias por tratar de sacar lo mejor de mí y por su compromiso con la academia. En especial por toda la ayuda brindada, dirección y respaldo durante todo este proceso.

DEDICATORIA

JOSÉ ALEJANDRO CAÑAR BENAVIDES

Dedico este logro, en primer lugar, a Dios a él sea toda gloria y toda honra. Porque él ha sido quien ha guiado mis pasos, me ha sostenido y me ha permitido llegar hasta aquí. Todo lo que soy, todo lo que tengo y todo lo que logre será gracias a él, a su infinito amor y misericordia.

A mis dos mamás, Silvia y Edith. A quienes han sido un pilar fundamental en mi vida, cada uno de sus esfuerzos, sacrificios y dedicación me ha traído a donde estoy el día de hoy y serán el pilar de donde estaré el día de mañana. Por ello, cada logro que obtenga será también el de ustedes.

Este logro lleva mi nombre, pero también el suyo.

Con todo mi amor y gratitud, se los dedico

AGRADECIMIENTO

VALERIA ABIGAIL IGLESIAS MUÑOZ

Me gustaría empezar agradeciendo a Dios por la oportunidad de llegar hasta aquí y por entregarme todas las bendiciones posibles para que pueda alcanzar mis metas. Le agradezco a Dios por poner en mi vida a todas las personas que tengo a mi lado y son mi soporte e inspiración para todo lo que me propongo, pero sobre todo le agradezco a Dios por su amor infinito y sus detalles en cada paso de mi vida universitaria, definitivamente todo lo que tengo y soy se lo debo a Dios.

Quiero también mencionar a mis padres, Francisco Iglesias y Sandrita Muñoz que son los mejores padres que uno puedo tener en su vida y a quienes les debo mucho más de lo que imaginan. Les agradezco su entrega y sacrificio para que yo pueda estar hoy aquí, también agradezco su amor y sus palabras de aliento que te impulsan a ser mejor, pero lo que más agradezco es su ejemplo para ser personas de bien y hacer los correcto siempre. Espero algún día poder devolverles todo lo que ustedes han hecho por mi y mucho más porque lo merecen.

A mis hermanos, Pablo y Franccesca, les agradezco ser mi mejor compañía y mi ejemplo para seguir en muchos aspectos de mi vida, gracias por los consejos en la vida universitaria, por las risas y por siempre ser mi red de apoyo cuando más lo necesito. Son el mejor regalo que tengo en mi vida. También quiero agradecerle a mi Kirita María que estuvo conmigo realizando este trabajo y como perrito fiel, me daba paz cuando lo necesitaba.

A mi novio, Naigel Ramón, le agradezco ser mi mejor amigo y compañero fiel en todo momento, gracias por tu amor y tu paciencia, por amarme incluso en los días en los que no podía conmigo misma. Gracias por tus consejos, por apoyarme y estar conmigo

en todo momento, por calmarme cuando sentía que no lo iba a lograr y poner de tu persona para ayudarte a concluir este trabajo con esfuerzo. Espero poder hacer los mismo por ti en su momento,143.

Le agradezco a mi compañero de tesis, José, por acompañarme en este proceso, no solo de la tesis sino en general, te agradezco por ser mi mejor amigo en esta carrera universitaria y por convertirte en un hermano, porque no solo estuviste en los momentos bueno, sino que estuviste en oración en los momentos más difíciles y en lo que más necesitaba ayuda. Gracias por tu amistad y compromiso con este trabajo. Quiero mencionar a mis amigos, Johnny, Ash, Zach y Ron en este agradecimiento por darme lo mejor de su amistad en esta etapa universitaria, los quiero y agradezco todos los momentos increíbles que vivimos como amigos, siempre los llevare en mi corazón.

Finalmente, quiero agradecer a nuestro tutor Félix Carrera, por su paciencia y tiempo para ayudarnos a realizar este trabajo con excelencia. Agradezco su dedicación y pasión por enseñar, el mundo necesita más profesores como usted, que compartan su pasión por una ciencia con sus alumnos, para así crear buenos profesionales.

DEDICATORIA

VALERIA ABIGAIL IGLESIAS MUÑOZ

Le dedico este trabajo y mi título, primero a Dios, porque para Él siempre será lo mejor de mí, todo mi honor y toda mi gloria serán siempre para Él.

También lo dedico a las personas que representan mucho en mi vida y a quienes ocupan un pedazo grande en mi corazón, a mis padres Francisco y Sandrita, a mis hermanos Pablo y Franccesca, a mi Kirita María y a mi Monchito. Son lo más valioso que tengo.

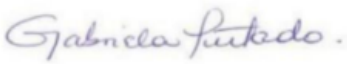
UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

TRIBUNAL DE SUSTENTACIÓN

f. _____

Ing. Carrera Buri, Félix Miguel, Mgs

TUTOR



f. _____

Ing. Hurtado Cevallos, Gabriela Elizabeth, Mgs

DECANO O DIRECTOR DE CARRERA

f. _____

Lcda. Zumba Córdova, Rosa Margarita PhD

COORDINADOR DEL ÁREA O DOCENTE DE LA CARRERA

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Este trabajo investiga la estimación de ventas de alimento balanceado para camarón mediante la combinación del algoritmo K-Means y un modelo de regresión lineal múltiple, bajo un enfoque cuantitativo, aplicado, descriptivo y predictivo. El estudio se realizó con una base histórica de 7.216 registros y 38 variables correspondientes a transacciones comerciales realizadas entre enero de 2025 y mayo de 2026. El procesamiento se realizó en Python mediante Google Colab, incluyendo inspección, limpieza, manejo de valores faltantes y atípicos, conversión de tipos, eliminación de variables irrelevantes, codificación One-Hot y estandarización. Se utilizó K-Means para realizar el agrupamiento y se evaluó el número óptimo de clústeres mediante el método del codo y el coeficiente de silueta, determinándose que la mejor configuración fue con dos grupos. Posteriormente, se incluyó la etiqueta Clúster como variable explicativa dentro de una regresión lineal múltiple con el fin de estimar Total USD. El modelo obtuvo un R^2 de 0,9883, un error cuadrático medio de 710.081,00 y un error absoluto medio de 485,65. Los resultados mostraron que los valores observados y estimados coincidieron en un alto grado, aunque se encontraron desviaciones mayores en las transacciones negativas o cercanas a cero. Se concluye que la combinación de K-Means y regresión lineal múltiple permite reconocer patrones transaccionales y generar estimaciones útiles para apoyar decisiones relacionadas con inventarios, compras, despachos, metas comerciales y recursos financieros, considerando las condiciones metodológicas específicas del modelo aplicado. La investigación también comparó sus resultados con estudios anteriores sobre segmentación y predicción comercial, encontrando coincidencias en la utilidad analítica de estas técnicas aplicadas.

Palabras clave: estimación de ventas, K-Means, machine learning, regresión lineal múltiple, alimento balanceado, sector camaronero.

ABSTRACT

This study investigates the estimation of shrimp feed sales through the combination of the K-Means algorithm and a multiple linear regression model, under a quantitative, applied, descriptive, and predictive approach. The study was conducted using a historical database containing 7,216 records and 38 variables corresponding to commercial transactions carried out between January 2025 and May 2026. Data processing was performed in Python through Google Colab, including inspection, cleaning, handling of missing and outlier values, data type conversion, removal of irrelevant variables, One-Hot encoding, and standardization. K-Means was used for clustering, and the optimal number of clusters was evaluated using the elbow method and the silhouette coefficient, determining that the best configuration consisted of two groups. Subsequently, the Cluster label was included as an explanatory variable within a multiple linear regression model in order to estimate Total USD. The model obtained an R^2 of 0.9883, a mean squared error of 710,081.00, and a mean absolute error of 485.65. The results showed a high degree of agreement between observed and estimated values, although greater deviations were found in negative transactions or those close to zero. It is concluded that the combination of K-Means and multiple linear regression makes it possible to identify transactional patterns and generate useful estimates to support decisions related to inventories, purchases, dispatches, sales targets, and financial resources, considering the specific methodological conditions of the applied model. The research also compared its results with previous studies on commercial segmentation and prediction, finding similarities in the analytical usefulness of these applied techniques.

Keywords: sales estimation, K-Means, machine learning, multiple linear regression, shrimp feed, shrimp sector.

RÉSUMÉ

Ce travail étudie l'estimation des ventes d'aliments équilibrés pour crevettes par la combinaison de l'algorithme K-Means et d'un modèle de régression linéaire multiple, selon une approche quantitative, appliquée, descriptive et prédictive. L'étude s'est appuyée sur une base historique de 7.216 transactions commerciales et 38 variables entre janvier 2025 et mai 2026. Le traitement en Python avec Google Colab comprenait l'inspection, le nettoyage, le traitement des valeurs manquantes et atypiques, la conversion des types, l'élimination des variables non pertinentes, l'encodage One-Hot et la standardisation. K-Means a été utilisé pour le regroupement, tandis que le nombre optimal de clusters a été évalué par la méthode du coude et le coefficient de silhouette, la meilleure configuration étant de deux groupes. Ensuite, l'étiquette Cluster a été intégrée comme variable explicative dans une régression linéaire multiple afin d'estimer Total USD. Le modèle a obtenu un R^2 de 0,9883, une erreur quadratique moyenne de 710.081,00 et une erreur absolue moyenne de 485,65. Les résultats ont montré une forte concordance entre les valeurs observées et estimées, malgré des écarts plus importants dans les transactions négatives ou proches de zéro. La combinaison de K-Means et de la régression linéaire multiple permet d'identifier des schémas transactionnels et de produire des estimations utiles pour les décisions concernant les stocks, les achats, les expéditions, les objectifs commerciaux et les ressources financières, selon les conditions méthodologiques du modèle. L'étude a comparé ses résultats aux recherches antérieures sur la segmentation et la prédiction commerciale, constatant des similitudes dans l'utilité analytique de ces techniques.

Mots-clés: estimation des ventes, K-Means, apprentissage automatique, régression linéaire multiple, aliment équilibré, secteur crevetticole.

1. INTRODUCCIÓN

La trayectoria del camarón como producto de uso humano se remonta hasta los asentamientos de comunidades de época antigua que habitaban en áreas costeras de Asia y del Mediterráneo, donde la captura del camarón ya formaba parte de las actividades de subsistencia de las comunidades de pesca tradicional. Con el desarrollo de la familia Penaeidae fue a través de los tipos de especies que actualmente dominan la acuicultura internacional. Fue a partir del siglo XX cuando la aplicabilidad económica de las especies mencionadas alcanzó cierta magnitud global, ya que la aparición de las técnicas de reproducción en cautividad cambió radicalmente la relación de la humanidad con el recurso marino. La ingesta de camarón por parte de las civilizaciones se da desde la antigua época en Asia y el Mediterráneo, sin embargo, a partir del siglo XX fue cuando se empezó a comercializar debido sobre todo a las técnicas de reproducción en cautividad (Stony Brook University, Florida International University, & National Taiwan Ocean University., 2016). La acuicultura del camarón acumuló volúmenes de producción de gran cuantía, esto fue incluso hasta mediados de la década de 1980 cuando la infraestructura productiva y el acceso a los circuitos de mercados internacionales sumándose a los avances en la biología de la reproducción permitieron generar las condiciones de desarrollo de una industria global (BM Editores, 2022).

El cultivo intensivo tuvo como primera referencia al Dr. Motosaku Fujinaga, quien en 1933 realizó experimentos de reproducción en cautiverio del Penaeus japonicus en Japón, dando lugar al cultivo comercial en 1955, tras más de una década de estudios sistemáticos. Después de varias décadas, tuvo lugar la primera reproducción artificial del Litopenaeus vannamei en Florida en 1973, a partir de una hembra silvestre de Panamá y, tras los resultados positivos obtenidos en estanques y el hallazgo de la ablación unilateral,

el cultivo de esta especie comenzó su proceso de expansión hacia Sudamérica y el área continental de EE. UU. Según las estadísticas de la FAO, la acuicultura de camarón alcanzó las 6.25 millones de toneladas en 2019, de las cuales el L. vannamei contribuyó al 87 %, cultivándose en más de 2 millones de hectáreas de estanques de países tropicales, siendo el 75 % en Asia y el resto en América Latina (BM Editores, 2022).

A lo largo de la historia la producción y comercialización de alimentos por parte de los humanos ha sido una constante sin embargo el crecimiento exponencial del que ha sufrido la población en el último tiempo ha hecho que la acuicultura se especialice cada vez más lo que ha provocado que ésta se enfoque hacia la productividad comercial con el fin de abastecer los mercados tanto nacionales como internacionales.

La hechura de la camaronicultura en América Latina se vio favorecida por condiciones ambientales muy propicias, por la existencia de zonas costeras estuarinas y por la creciente demanda de divisas de economías en vías de desarrollo. La región aglutina aproximadamente un tercio de las exportaciones mundiales de camarón, a la cabeza del ranking se encuentra Ecuador, seguido por Argentina y Honduras (Lupa, 2025). La elaboración del alimento balanceado llegó a ser uno de los costes de explotación más altos, pues en algunas de las camaroneras llegaba a alcanzar niveles casi del 60% sobre los costes totales, lo que ya desde las primeras etapas dejaba entrever la necesidad del desarrollo de una cadena de suministros de insumos nutricionales fuerte y eficiente en la región.

El caso de Ecuador constituye uno de los relatos más únicos de la acuicultura global. La industria camaronera ecuatoriana se originó casualmente en 1968 cuando, en el archipiélago de Jambelí en el cantón de Santa Rosa, el ascenso de la marea llevó a los camarones a los salitrales de los manglares adyacentes, donde quedaron atrapados,

crecieron y mostraron ese potencial productivo que quería demostrar el texto de las lagunas costeras ecuatorianas (Sustainable Shrimp Partnership [SSP], 2023).

Durante la década de los 80 se establecieron cerca de 90000 hectáreas destinadas a la producción del camarón mientras que para el año 1995 ya existían alrededor de 1800 hectáreas (Global Seafood Advocate, 2018). Pero es de esto la industria camaronera se ubicó como una de las fuentes de exportación más importantes del país por detrás del petróleo, para el año 2022 el sector camaronero creció en un 37% en comparación al año anterior (REVISTVR, 2025). Para el año 2025 el sector camaronero pasó a ser el principal sector exportador del país por delante del petróleo al haber exportado 4254 de dólares (Primicias, 2025).

La producción en expansión exigía cada vez más suministro de alimento balanceado de calidad. La investigación de los requerimientos nutricionales de penaeidos empezó a partir de 1970 e implicó la creación de una extensa industria de fabricación de alimentos que asistió no solo a la nutrición y fisiología del animal, sino también a la tecnología de manufactura. La intensificación del cultivo llegó con requerimientos de dietas con mayores densidades nutricionales, que eran propias en el momento de maximizar el rendimiento que podían garantizarse en las condiciones de alta densidad de cultivo, mayor biomasa y uso intensivo de tecnologías de aireación y alimentación automática El constante crecimiento de este sector evidenció la necesidad de realizar estudios en cuanto a la nutrición que requerían los camarones para favorecer su desarrollo (Confirmado, 2026).

En este marco, las empresas distribuidoras de alimento balanceado para camarón se enfrentaron a la necesidad de desarrollar procesos de planificación comercial que fueran capaces de prever la demanda con suficiente antelación a las fechas de

planificación de la producción, el almacenamiento y la distribución, al tiempo que se lleva a cabo el proceso de aprovisionamiento propio del negocio. A partir de la imprevisibilidad del ciclo productivo del camarón y de los factores asociados a la densidad de siembra, las fases de crecimiento y la recolección a la que se aspira, los sistemas de proyección tradicional mediante series históricas simples y los juicios de los expertos comerciales resultan insuficientes para alcanzar la complejidad estructural de la realidad de la demanda real. Adicionalmente, la relevancia económica que se otorga al sector camarón vuelve necesario utilizar, y por tanto, aprovechar todos los datos posibles, junto con herramientas estadísticas usadas en ciencias de la computación para, en su conjunto, permitir una adecuada toma de decisiones estratégicas ante la incertidumbre que cobra protagonismo en el entorno competitivo (Redalyc, 2022).

A lo largo de los años se ha ido creando diversas técnicas de machine learning esta herramienta tuvo sus inicios en América del sur en el siglo 21 gracias sobre todo al acceso a la computación (Suyal & Sharma, A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning, 2024). Esta herramienta ha permitido a las empresas entrar en información relevante de una gran base de datos con las cuales les ha permitido tomar las mejores decisiones que ayuden a optimizar los procesos productivos. De acuerdo con Lorente-Leyva et al. (2020) a la hora de predecir los resultados los algoritmos de machine learning son más eficientes que los métodos tradicionales. En el Ecuador se suele implementar modelos híbridos de machine learning con el fin de poder pronosticar las ventas y realizar predicciones financieras (Redalyc, 2022)

En Ecuador, la madurez del sector camaronero y la disponibilidad de medios de cómputo modernos han propiciado la posibilidad de implementar modelos híbridos de machine learning para la gestión comercial de empresas pertenecientes a la cadena de

valor del camarón, siendo el proceso mediante el cual se implementan algoritmos supervisados para modelar pronósticos de ventas de comercializadoras ecuatorianas de camarón y confrontarlas con proyecciones financieras uno de esos caminos de investigación novedosos con potencial de transformación para el sector (Redalyc, 2022).

El siguiente trabajo combina el algoritmo K-Means para la segmentación no supervisada de patrones de demanda con un modelo de regresión lineal múltiple generalizado al conjunto de transacciones analizadas. Los clústeres que resultan de aplicar el algoritmo K-Means se añaden posteriormente al modelo predictivo como una variable explicativa más, con el objetivo de representar las diferencias identificadas entre los grupos dentro de una única estimación de ventas (Hartomo & Nataliani, 2021).

2. PROBLEMÁTICA

La importancia que ha tenido en los últimos años del sector acuícola ha provocado que las empresas dedicadas a la comercialización del camarón deban coordinar de manera simultánea varios elementos importantes tanto para la producción como para la comercialización como pueden ser las ventas, inventarios, abastecimiento, entre otras cosas, por lo que todo debe estar bien estimado o medido para no afectar el comportamiento de las operaciones

Una de las dificultades que se suele presentar a la hora de estimar las ventas es la variabilidad de ciertos aspectos como el precio la ubicación del cliente la cantidad del producto requerido entre otras cosas, así también existen factores biológicos y ambientales que pueden alterar la estimación. Gladju et al. (2022), menciona que para poder estimar estas variables es necesario el empleo de minería de datos y machine learning.

En la empresa analizada en este estudio se evidencia la necesidad de implementar machine learning ya que existen 7216 transacciones, las cuales se dividen en 38 variables relativas a los clientes, productos cantidades, precio, entre otros lo que imposibilita poder mantener un control manual adecuado y poder clasificar estas transacciones en grupos comunes

Ante esta problemática el K-Means se presenta como una metodología capaz de agrupar a las transacciones comunes y ordenarlas en base a variables como productos volúmenes, precios entre otros, Para ello es necesario que el algoritmo cuente con variables comparables, así como también con una adecuada codificación y el establecimiento de un número determinado de grupos.

El problema principal es la falta de documentación que ayude a identificar patrones y estimar valores comerciales, además la gran cantidad de variables con las que se cuenta hace que la revisión manual sea ineficiente, por lo que ante esta situación la pregunta de investigación planteada es la siguiente ¿de qué manera la aplicación del algoritmo K-Means, junto con un modelo de regresión lineal múltiple, permite agrupar las transacciones históricas y estimar el valor de las ventas de una empresa comercializadora de alimento balanceado para camarón?

3. JUSTIFICACIÓN

Esta investigación se justifica por la necesidad de realizar una adecuada estimación de ventas del balanceado para camarón, ya que este sector es una parte importante de la cadena productiva de la acuicultura. Durante el año 2025 la exportación de camarón se convirtió en la actividad más importante dentro del país por encima del petróleo, lo que evidencia la importancia que tiene este sector en la economía ecuatoriana (Primicias, 2025).

Además, la elección de K-Means en esta investigación se justifica por su capacidad para clasificar observaciones similares, ya sea en base a variables como volumen, precio, descuento, producto, fecha, lugar y tipo de operación, así también la elección de la regresión lineal múltiple se justifica por la necesidad de estimar cada variable y determinar la dirección que esta toma en el futuro (James, 2023)

La justificación desde el punto de vista metodológico por el tratamiento dado a los datos previos a la ejecución del modelo ya que en ciertas empresas pueden existir campos vacíos de las variables o valores atípicos los cuales deben ser tratados de manera especial por ello para el presente estudio se realizó una limpieza de los datos para posteriormente ser seleccionados y codificados antes de aplicar la metodología -Means y regresión lineal múltiple.

En cuanto a la justificación práctica esta investigación permite a los gerentes de las empresas poder tomar decisiones en cuanto a la organizar inventarios, establecer objetivos, coordinar diferentes áreas de la empresa, establecer indicadores entre otras cosas que conlleven a un mejor funcionamiento de la parte operativa de la empresa.

Desde lo académico la presente investigación vincula técnicas supervisadas y no supervisadas a un problema empresarial en el Ecuador permitiendo comprender como el tratamiento de los datos influyen en la capacidad de obtener datos fiables, así también el presente estudio dota de la capacidad de análisis predictivo con el cual tomar las mejores decisiones en el ámbito empresarial.

4. ALCANCE

El presente trabajo de titulación con el título de "Estimación de ventas mediante un algoritmo de k means para la predicción basados en machine learning." tiene como objetivo analizar el comportamiento histórico de las ventas en el sector acuícola mediante el uso de técnicas de aprendizaje automático. La investigación se enfoca en la identificación de patrones y agrupaciones presentes en los datos comerciales, utilizando el algoritmo K-means como herramienta de segmentación, con la finalidad de generar estimaciones de ventas que contribuyan a mejorar la planificación comercial y la toma de decisiones empresariales.

Esta investigación está dirigida principalmente a tres grupos de interés. Por un lado, los responsables de la gestión comercial y de ventas quienes podrán utilizar los resultados obtenidos para identificar patrones de comportamiento en las ventas, mejorar la planificación de la demanda y establecer estrategias comerciales basadas en el análisis de datos.

Por otro lado, los responsables de la planificación y gestión de la producción podrán utilizar las estimaciones generadas para anticipar posibles variaciones en la

demanda, optimizar la planificación de los recursos y reducir los riesgos asociados con la sobreproducción o la insuficiencia de inventarios.

Finalmente, los directivos y responsables de la toma de decisiones en empresas del sector acuícola podrán contar con información basada en técnicas de Machine Learning que les permita evaluar de manera más precisa el comportamiento esperado de las ventas y respaldar sus decisiones estratégicas mediante información cuantitativa y patrones identificados a partir de datos históricos.

5. OBJETIVOS

5.1. Objetivo General

Analizar la estimación de ventas de alimento balanceado para camarón mediante un algoritmo de K-Means para la predicción, basados en Machine Learning.

5.2. Objetivos Específicos

●Indagar en la literatura existente sobre las tendencias de venta dentro del sector acuícola, así como los distintos enfoques de Machine Learning aplicables a este tipo de análisis.

● Utilizar modelos fundamentados en técnicas de Machine Learning para procesar y analizar la información histórica de ventas de una empresa dedicada a la comercialización de balanceado para camarón.

● Contrastar los resultados alcanzados frente a investigaciones anteriores que hayan aplicado metodologías afines de segmentación y estimación de ventas.

6. MARCO TEÓRICO

6.1. Inteligencia Artificial

En los últimos años la inteligencia artificial (IA) ha mantenido un crecimiento exponencial, de acuerdo con Russell y Norvig (2021) la IA es una parte de la informática dedicada a la automatización, razonamiento y el aprendizaje en continuo, con estas herramientas las empresas han podido procesar la información de manera automática y veloz y tomar decisiones estratégicas en base a la información procesada.

La IA se ha desarrollado en base computacionales que superan las facultades cognitivas humanas, estos modelos son capaces de procesar una gran cantidad de datos, determinar patrones y poder realizar predicciones o recomendaciones avanzadas, la inteligencia artificial en los últimos años ha sido empleada en diferentes sectores como la salud, las finanzas, la logística, así como también en procesos de producción y comercialización (Sarker, 2021).

6.1.1. Inteligencia Artificial en el Sector Acuícola

El sector acuícola ha sido 1 de los pioneros en emplear la inteligencia artificial debido a su gran crecimiento en los últimos años, de acuerdo con Russell y Norvig (2021) herramientas como computed vision, machine learning o predictive modeling que están basadas en inteligencia artificial, han permitido que el sector acuícola mejora su eficiencia y productividad, ya que les ha permitido mantener un control eficiente sobre las posibles enfermedades así como también mantener una adecuada gestión medioambiental y optimizar los procesos productivos

Dentro del sector de la venta de balanceados de acuícolas la inteligencia artificial dota a las empresas de una cierta ventaja competitiva ya que les permite predecir la

demanda futura, segmentar a los clientes, realizar proyecciones de ventas entre otras cosas. Rabia et al. (2024) establece que nuevas tecnologías basadas en inteligencia artificial han sido implementadas en grandes y pequeñas granjas acuícolas con el fin de incrementar su productividad

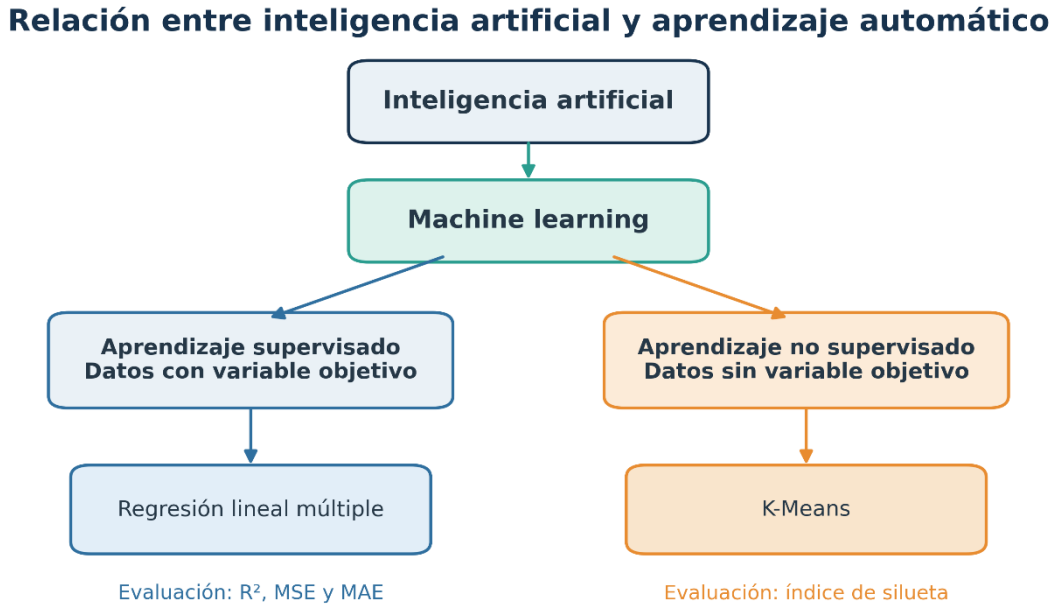
6.1.2. Machine Learning

El machine learning es una parte de la inteligencia artificial, el cual es capaz aprender en base a lo ya experimentado. Con respecto a esto Sarker (2021) menciona que el machine learning busca realizar predicciones en base a datos no observados y de esta manera poder tomar las mejores decisiones

El proceso de aprendizaje se basa en la entrada y salida de información con la cual el algoritmo ajusta los parámetros para reducir los errores de predicción. Para ello, cambia ciertos valores internos con el objetivo de precisar mejor los resultados. La calidad del machine learning depende de los datos disponibles y de la calidad que estos posean (Russell y Norvig, 2021).

Figura 1

Relación entre inteligencia artificial y aprendizaje automático



Elaboración propia con base en Russell y Norvig (2021) y James et al. (2023).

6.1.3. Machine Learning en Acuicultura y Sector Camaronero

La implementación del marching learning en la acuicultura se ha expandido en los últimos años debido sobre todo a la gran expansión que ha tenido este sector. Aung et al. (2025), menciona tras su análisis que las inteligencias artificiales más implementadas en el sector de la acuicultura han sido el aprendizaje automático supervisado y los sistemas de visión artificial con las cuales se hacen predicciones acerca de la calidad del agua, el tipo de enfermedades, entre otras ventajas más como la optimización de estrategias de alimentación.

La implementación del margin learning dentro de las empresas dedicadas a la comercialización de balanceado para camarón ayuda a que estas puedan comprender el comportamiento de los clientes y de esta manera poder proyectar el volumen de ventas así como también las dinámicas del ciclo productivo del camarón. Los productores camaroneros adquieren alimentos en base a las etapas de crecimiento y expectativas de

cosecha por lo que la realización de proyecciones mediante algoritmos de machine learning se hace indispensable (Aung, Abdul Razak, & Rahiman Bin Md Nor, 2025).

6.1.4. Modelos Supervisados de Machine Learning

Los modelos supervisados son una de las categorías básicas que engloba el machine learning y que se distingue por aprender a partir de un conjunto de datos etiquetado, es decir, un conjunto en el que, para cada observación de entrada, existe a su vez una variable de salida conocida. Tal y como definen Sathya y Abraham (2013), el aprendizaje supervisado es aquel en el que recibe pares de entrada y salida en el transcurso del entrenamiento con la finalidad de adquirir una función de mapeo que vincule las variables predictoras con la variable objetivo, tal que pueda generalizar dicha relación ante nuevas observaciones no conocidas. Los datos de entrada (features) con los datos de etiqueta (labels) son aportados a la vez al modelo durante la fase de aprendizaje.

Aplicaciones de Modelos Supervisados en Estimación de Ventas

A lo largo de los últimos años se han realizado varios estudios en relación a la estimación de ventas. Malik et al. (2024), establece que los modelos de aprendizaje automático que son supervisados son más eficientes que aquellos modelos empleados tradicionalmente, ya que el aprendizaje automático analiza de manera simultánea varios elementos y es capaz también de reconocer patrones que son complejos de complejos de determinar, además menciona que para conocer la efectividad del modelo es necesario tomar en cuenta medidas como R^2, RMSE y MAE ya que estas permiten comprar los resultados.

En el contexto sudamericano también ha habido investigaciones relacionadas con esta temática, donde Lorente-Leyva et al. (2020) en un estudio realizado en el sector textil, manifiesta que el machine learning supera en precisión y rendimiento a los modelos

tradicionales. Estos resultados evidencian la importancia de implementar modelos supervisados dentro de las empresas en comparación a modelos tradicionales que son deficientes.

Evaluación de Modelos Supervisados para la predicción

Para evaluar las predicciones de un modelo es necesario emplear métricas estadísticas con la cual comparar los resultados obtenidos con aquellos que son reales, en dónde R^2 manifiesta que también es explicado los datos por parte del modelo, mientras que el el RMSE muestra el nivel de error que éste ha tenido, finalmente él MAE indica la cantidad de veces de un modelo se equivoca en promedio, estas métricas no siempre son las mismas ya que deben ser escogidas el problema que se requiere analizar y de la importancia que se le dé a los errores en la toma de decisiones.

Modelos No Supervisados de Machine Learning

Los modelos no supervisados constituyen la segunda de las grandes categorías del machine learning y se diferencian de los supervisados en que trabaja sobre datos no etiquetados, es decir, sobre conjuntos que no tienen una variable de salida predefinida. Según Suyal y Sharma (2024), establece que en este aprendizaje la salida de información se realiza en base a búsqueda de similitudes, además mencionan que este tipo de modelos son útiles cuando no se disponen de datos históricos etiquetados.

6.1.5. Clustering

El clustering es una de las técnicas más empleadas cuando se tiene datos comerciales y científicos el principio básico de este es agrupar datos semejantes entre si en un cluster (Suyal & Sharma, 2024). Dentro de las empresas dedicadas de la distribución de alimento del balanceado el clustering les permite segmentar a los clientes

en base a patrones de compra parecidos, además les permite determinar estacionalidades de la demanda, entre otras clasificaciones con las cuales poder hacer proyecciones en base a las particularidades de cada cluster (Aulia, Priatna, & Yasir, 2024).

6.1.6. Modelos No Supervisados en el Sector Acuícola

Los métodos no supervisados dentro del sector de la acuicultura han sido ampliamente estudiados por varios autores, entre ellos Islam et al. (2024), quienes proponen la implementación IoT y machine learning para el control de granjas camaroneras en Bangladesh, así como también para predecir condiciones ambientales desfavorables sin la etiqueta de los datos. Su estudio demostró que la implementación de estas técnicas tiene una alta capacidad de detección de patrones con las cuales se puede tomar las mejores decisiones y formular estrategias de alerta y prevención.

6.1.7. Algoritmo K-Means

El K-Means es uno de los métodos de agrupamiento más empleados debido a su eficiencia y versatilidad en diferentes áreas. Suyal y Sharma (2024) mencionan que este método es un algoritmo basado en el aprendizaje no supervisado, en donde los datos son agrupados en cluster en base a características similares, y aquellos datos que no mantienen similitud son distribuidos en diferentes clusters lo que garantiza la reproductividad del resultado final.

6.1.8. Fundamento Matemático del K-Means

De forma precisa, el método K-Means se define como una técnica que busca una división $C = \{C1, C2, \ldots, Ck\}$ del conjunto de datos X en k grupos disjuntos de datos que minimice su función objetivo: la inercia o la suma de los cuadrados intra-cluster (Within-Cluster Sum of Squares, WCSS). Esta última se define como la suma de las distancias

euclidianas al cuadrado entre cada uno de los puntos xi y el centroide mui del cluster asignado. La minimización de esta función asegura que los grupos encontrados sean homogéneos dentro de los mismos, heterogéneos entre ellos, es decir, que los puntos dentro de un mismo grupo sean lo más parecidos entre sí y, en cambio, lo más diferentes posible respecto de los puntos de otros grupos (Suyal & Sharma, 2024).

6.1.9. Determinación del Número Óptimo de Clusters

Establecer el numero k de clusters es fundamental antes de implementar el algoritmo K-Means, al no haber un valor oprimo, el método del codo es uno de los mas empleados para determinar este valor, para ello es necesario graficar primeramente todos lo valores de WCSS y seleccionar el punto de inflexión donde la reducción de inercia por la entrada de un nuevo cluster se vuelve marginal.

6.1.10. Índice de Silueta (Silhouette Score)

Rousseeuw (1987) propuso el índice de silueta como una medida interna destinada a examinar simultáneamente la cohesión y la separación de los grupos obtenidos mediante un algoritmo de agrupamiento.

Para cada observación i, el índice de silueta se obtiene mediante la siguiente expresión matemática:

$$s(i) = [b(i) - a(i)]/maxa(i), b(i)$$

El término a(i) corresponde a la distancia media entre la observación y los demás elementos pertenecientes a su propio clúster. El término b(i) representa la menor distancia media entre dicha observación y los elementos de cualquier clúster diferente. La puntuación general de una partición se calcula mediante el promedio de los valores individuales obtenidos para todas las observaciones analizadas (Rousseeuw, 1987).

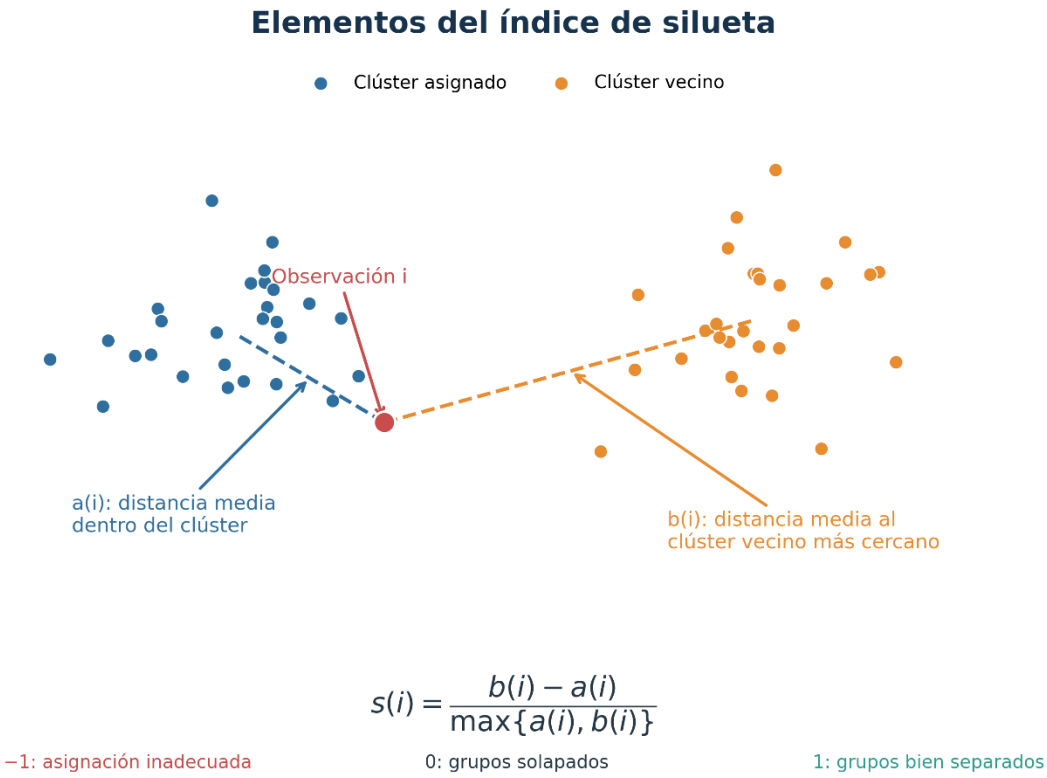
Los resultados asociados al índice se encuentran dentro del intervalo comprendido entre −1 y 1. Los valores cercanos a 1 muestran agrupaciones compactas entre las observaciones analizadas y claramente separadas de los grupos próximos. Las puntuaciones que se hallan cercanas a 0 muestran observaciones situadas en límites poco definidos entre dos agrupaciones. Los valores negativos indican que una observación presenta una menor distancia promedio respecto de una agrupación diferente de aquella a la que fue asignada.

El índice de silueta ha de ser comparado entre las soluciones que se obtienen con las mismas variables, escalas, distancias y procedimientos de preprocesamiento aplicados. No existe un umbral universal que determine por sí mismo la validez sustantiva de una segmentación, porque la estructura de los datos condiciona el valor alcanzado. Su aplicación tampoco reemplaza la interpretación conceptual de los grupos ni la revisión de tamaños reducidos, valores atípicos o agrupaciones carentes de utilidad práctica (Scikit-learn developers, 2026b).

La selección del número de clústeres puede apoyarse en la solución que presente el mayor promedio de silueta entre varios valores admisibles de k. Esta comparación debe complementarse con el método del codo, la estabilidad de las asignaciones y la interpretación de los perfiles obtenidos. Una decisión sustentada en varios criterios disminuye el riesgo de escoger agrupaciones matemáticamente compactas que carezcan de significado dentro del dominio analizado.

Figura 2

Distancias empleadas para calcular el índice de silueta



Elaboración propia con base en Rousseeuw (1987).

6.1.11. K-Means en la Estimación de Ventas y el Sector Productivo

En la literatura científica reciente queda por validar a través de la validación empírica la combinación del K-Means con modelos de regresión para estimación de ventas. Sutara et al. (2024), en un artículo publicado en el International Journal of Information Technology and Computer Science Applications, aplicaron el algoritmo K-Means segmentación de clientes de acuerdo con la frecuencia de compra y el valor total de las transacciones, encontrando tres segmentos diferentes, con un coeficiente de silueta de 0,7913. Posteriormente aplicaron regresión lineal simple sobre cada uno de los segmentos, encontrando un coeficiente de determinación R^2 de 0.6910 para poder predecir las ventas, un resultado que demuestra que el modelo híbrido tiene potencial para encontrar estructuras latentes en los datos de transacciones comerciales, y mejorar la capacidad predictiva de los modelos de regresión que lo siguen.

En lo que respecta a la fundamentación metodológica, Hartomo y Nataliani (2021), en su trabajo de la banca de datos PMC, idearon un modelo que predecía utilizando el K-Means junto con algoritmos de series de tiempo para cada uno de los clusters, donde se probó, con datos empíricos, que el modelo híbrido se comportaba de forma significativamente superior al modelo de pronóstico, que no realiza la segmentación de forma previa. El hallazgo de estos autores da cobertura al uso del K-Means para hacer una primera etapa en el desarrollo de modelos predictivos de ventas, en el sentido de que cada segmento pueda ser modelado utilizando la técnica supervisada más adecuada en función de los patrones de comportamiento que tenga definido el segmento.

De cara al caso de una empresa comercializadora de alimento balanceado para camarón, la técnica del K-Means permite segmentar el histórico de ventas en función de variables como el volumen mensual vendido, la frecuencia de pedidos por cliente, la etapa del ciclo productivo del camarón, la zona geográfica de las granjas atendidas por las empresas y el tipo de producto adquirido por la clientela. Al hacerlo, se pueden identificar patrones de demanda diferenciados que podrían no detectarse en la visión agregada y, a partir de la segmentación obtenida, incorporar la pertenencia a cada clúster como información adicional dentro de un modelo general de regresión aplicado al conjunto de observaciones (Aulia, Priatna, & Yasir, 2024).

6.1.12. Modelos de Regresión en Machine Learning

La regresión es la familia de modelos de machine learning en modalidad supervisada cuyo objetivo es predecir el valor de una variable continua, el valor de una variable dependiente determinado a partir de un conjunto de variables predictoras. A diferencia de los modelos de clasificación, cuyo objetivo es asignar observaciones a una

categoría discreta, los modelos de regresión proporcionan estimaciones en términos numéricos que permiten cuantificar la magnitud de la variable a predecir, por lo que se convierten en herramientas especialmente adecuadas para la predicción de las ventas, ingresos, demanda u otros indicadores económicos expresados en términos continuos. La flexibilidad de los modelos de regresión permite incluir desde relaciones lineales sencillas hasta interacciones no lineales complejas entre las variables, en función del algoritmo usado (Sarker, 2021).

6.1.13. Regresión Lineal

La regresión lineal es el modelo de regresión por excelencia y es el fundamento conceptual sobre el cual las aproximaciones más complejas encuentran soporte; este modelo en concreto establece la relación lineal entre los predictores y la respuesta, y está descrito por una función afín cuyos coeficientes son estimados mediante el método de mínimos cuadrados ordinarios (OLS) (Ordinary Least Squares). La regresión lineal simple tiene en cuenta un único predictor y la regresión lineal múltiple extiende tal modelo a múltiples predictores simultáneamente, permitiendo la consideración de la influencia conjunta de varios predictores explicativos con respecto a la variable de interés (Sathya & Abraham, 2013).

La aplicabilidad de la regresión lineal a la estimación de ventas se encuentra ampliamente documentada en la literatura científica. Sutara et al. (2024) indicaron un coeficiente de determinación R^2 de 0.6910 al aplicar regresión lineal simple a datos de ventas transaccionales, lo cual refleja una capacidad predictiva adecuada para apoyar la toma de decisiones y ajustar las estimaciones de ingresos esperados. Estos autores determinan que la regresión lineal simple aplicada después de la segmentación mediante

K-Means, sobre subgrupos de datos con mayor homogeneidad interna, presenta un mayor potencial para representar el comportamiento de compra.

6.1.14. Modelos de Regresión Avanzados

Más allá de aplicar la técnica de regresión lineal, la literatura científica actual también ha explorado modelos de regresión más complejos para la previsión de ventas. En un contexto empírico publicado en Springer, los autores de Olaitan et al. (2024) encontraron que el modelo XGBoost fue superior a todos los algoritmos de predicción probados, seguido muy de cerca de la regresión lineal mediante un procedimiento de evaluación con MAE, RMSE y R^2. Este resultado es indicativo de que los modelos de ensamble pueden aportar conclusiones más satisfactorias cuando se trabaja con datos complejos y heterogéneos, pero la regresión lineal es competitiva bajo la condicional de que previamente hayan segmentado los datos a fin de minimizar su variabilidad interna.

La investigación de Malik et al. (2024) en el sector del retail muestra que los modelos supervisados de machine learning, que incluyen la regresión lineal y la regresión con bosques aleatorios (Random Forest Regression), presentan una mayor capacidad para capturar la complejidad dinámica de las predicciones de ventas que los métodos tradicionales de series de tiempo, como el modelo ARIMA. De hecho, los autores de Malik et al. (2024) concluyeron que la regresión lineal tiene una solidez estadística que es superior al modelo de series de tiempo, en el caso de que la intención del modelo objetivo sea mostrar la complejidad de la base de datos de ventas de entornos con variables que son contextualmente confusas.

6.1.15. Modelo Híbrido K-Means y Regresión

El modelo que se obtiene de la hibridación de K-Means como etapa de segmentación no supervisada con modelos de regresión como etapa predictiva

supervisada representa un marco metodológico sólidamente respaldado por la literatura especializada. Por medio de un estudio publicado en PMC, Hartomo y Nataliani (2021) demostraron que la combinación de algoritmos de clustering con modelos de pronóstico específicos por grupo produce predicciones más precisas y estables que los modelos globales aplicados sobre el conjunto global de datos sin segmentar. El principio en el que se basa la obtención de este modelo híbrido es que al reducir la heterogeneidad interna de los datos por medio del clustering previo, los modelos de regresión son capaces de describir con mayor fidelidad las relaciones entre las variables predictoras y la variable de respuesta para cada grupo diferencial.

Este planteamiento se encuentra en sintonía con el que desarrollaron Majeed et al. (2025), que reflejan que los marcos de arquitectura híbrida que combinan en primera instancia clustering no supervisado seguido de modelos supervisados específicos por cluster permiten lograr mejoras sistemáticas en la precisión gracias a las distintas aproximaciones que no segmentan. En el caso cambiante de la predicción de ventas de balanceados para camarón, en el que la heterogeneidad de los patrones de la demanda entre los distintos tipos de clientes y las diferentes zonas geográficas y fases del ciclo productivo requieren de una segmentación previa con K-Means especialmente adecuada y metodológicamente justificada.

Específicamente para la compañía estudiada, el modelo híbrido K-Means y regresión propone segmentar el histórico de venta en grupos con patrones de la demanda internamente suficientemente similares y externamente suficientemente diferentes, sobre los que se aplican las ecuaciones de regresión ajustadas a la naturaleza del segmento. El resultado que se genera es un sistema de predicción de la venta que aprovecha las potencialidades del aprendizaje no supervisado para estructurar los datos, y del aprendizaje supervisado para contribuir a la generación de predicciones cuantitativas más

que satisfactorias y que superan a los modelos aproximativos muy específicos de la historia de las ventas o las aproximaciones basadas en el juicio experto del grupo comercial (Aulia, Priatna, & Yasir, 2024; Hartomo & Nataliani, 2021).

6.2. Ecuaciones Matemáticas del Modelo K-Means

La formalización del algoritmo K-Means desde una perspectiva matemática constituyen un de los pilares principales para conocer su funcionamiento y valorar su uso en la segmentación de los datos comerciales. La función objetivo del algoritmo K-Means, que se denomina inercia o suma de cuadrados intra-cluster (Within-Cluster Sum of Squares, WCSS), tiene la siguiente expresión: se trata de una función que expresa la dispersión total de las observaciones respecto de los centroides de cada uno de los grupos obtenidos mediante este algoritmo de clustering y cuya minimización va a permitir la consolidación de la cohesión interna de cada uno de los grupos resultantes (Suyal & Sharma, A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning, 2024):

$$\boldsymbol{WCSS} = \sum \sum ||\boldsymbol{x}_{\mathrm{i}} - \boldsymbol{\mu}_{\mathrm{k}}||^{2}{}_{2}$$

Donde cada x_i corresponde a cada observación que contiene el conjunto de datos, μ_k es el centroide del cluster k al que corresponde la observación específica x_i, y $||\cdot||$ denota la norma euclidiana. La sumatoria exterior va recogiendo los grupos k que hemos definido, al paso que la sumatoria interior recoge la suma de las distancias cuadráticas entre cada punto que pertenece a dicho cluster y a su centroide. De este modo, la minimización de la función se asegura que los grupos que obtenemos son homogéneos entre sí y heterogéneos entre grupos diferentes (Sarker I. H., 2021).

La fase de asignación del algoritmo, y la cual constituirá el primer paso de cada paso iterativo de dicho algoritmo, se encuentra formalizada mediante la siguiente expresión, en donde cada observación recibirá la asignación de cluster que sea la que implique la minimización de la distancia euclidiana a la observación x_i, garantizando que cada punto x_i pertenezca solo al cluster responsable de producir la distancia al centroide más corto en el espacio de características:

$$\boldsymbol{C}(\boldsymbol{x}_i) = \boldsymbol{argmin}_k ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^{2_2}$$

La fase de actualización de los centroides, que constituye el segundo paso de cada iteración, se expresa mediante el cálculo de la media aritmética de todas las observaciones asignadas a cada cluster en la iteración precedente, actualizando la posición del centroide con base en la composición actual del grupo:

$$\mu_k^{(t+1)} = \frac{1}{\left|C_k^{(t)}\right|} \sum_{x_i \in C_k^{(t)}} x_i$$

Donde $|C_k|$ corresponde al número de observaciones asignadas al cluster k en esta iteración, y la sumatoria abarca todas las observaciones x_i que pertenecen al cluster. El proceso iterativo entre la fase de asignación y la fase de actualización se repite hasta que se logre la convergencia, condición que se da cuando no ha cambiado ninguna observación de cluster entre iteraciones sucesivas, condición que garantiza que se ha llegado a un mínimo local de la función WCSS (Suyal & Sharma, A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning, 2024).

6.3. Ecuación de Regresión Lineal Múltiple

La regresión lineal, tanto en su forma simple como en la múltiple, tiene como base una ecuación algebraica que formaliza la relación funcional entre el conjunto de variables predictoras y la variable de respuesta. La forma general de la ecuación de regresión lineal múltiple, empleada para la estimación de ventas a partir de una condición de múltiples variables explicativas del ciclo productivo camaronero, se describe como sigue (Sathya & Abraham, 2013):

$$\hat{y} = \boldsymbol{\beta^0} + \boldsymbol{\beta^1 x^1} + \boldsymbol{\beta^2 x^2} + \ldots + \boldsymbol{\beta}_n \boldsymbol{x}_n + \boldsymbol{\varepsilon}$$

n donde $\hat{y}$ representa el valor pronosticado de la variable dependiente (volumen de ventas de alimento balanceado); β_0 corresponde al término intercepto u ordenada del modelo; β_1, β_2, …, β_n son los coeficientes de regresión simple, que expresan el efecto marginal de cada una de las variables predictivas x_1, x_2, …, x_n sobre la variable de respuesta, considerando las demás variables explicativas del modelo como constantes, y ε equivale al término de error aleatorio (debido a otras variables que no aparecen en el modelo y cuya variabilidad es explicada por las mismas) (Malik, Khan, Abid, & Aslam, 2024).

Los coeficientes β han sido estimados por el método de mínimos cuadrados ordinarios (OLS o Ordinary Least Squares), que procura obtener aquella mínima suma de cuadrados de los residuos entre los valores predichos y los valores de la variable dependiente observada o medida, la cual puede ser expresada como: min $\sum(y_i - \hat{y}_i)^2$ (Sarker I. H., 2021).

6.3.1. Coeficiente de Determinación (R^2)

Chicco et al. (2021) definen el coeficiente de determinación como una medida relativa del ajuste alcanzado por un modelo de regresión respecto de la variabilidad total observada en la variable de respuesta. Este indicador compara los errores residuales del modelo con las desviaciones que presentan los valores observados respecto de su media aritmética. Por esta razón, su lectura expresa cuánto mejora la regresión frente a una predicción de referencia que asignaría la media a todas las observaciones.

El coeficiente de determinación se calcula mediante la siguiente expresión matemática:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^2}$$

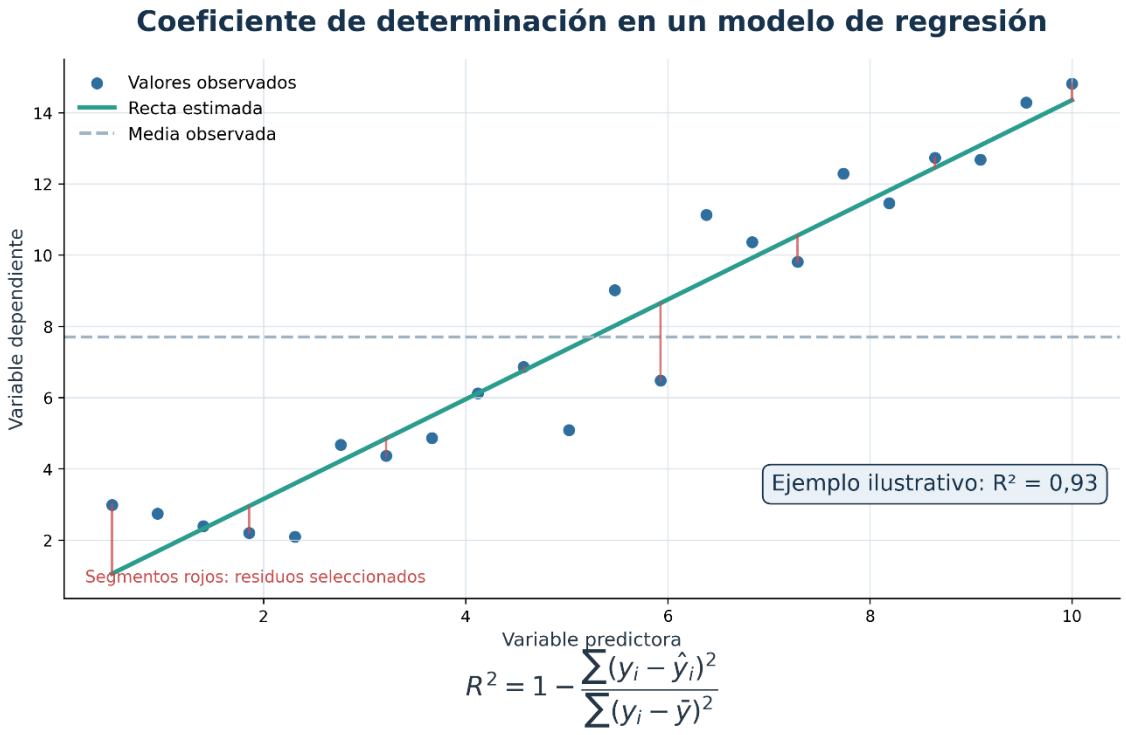
En esta expresión, y_i representa el valor observado de la variable dependiente, mientras $\hat{y}_i$ corresponde al valor estimado por el modelo para la misma observación. El término $\bar{y}$ representa la media de los valores observados dentro del conjunto utilizado para evaluar la regresión. El numerador cuantifica la suma de cuadrados de los residuos, mientras el denominador representa la suma total de cuadrados respecto de la media observada (Montgomery et al., 2021).

Un valor de R^2 igual a 1 representa una coincidencia perfecta entre los valores observados y las estimaciones producidas por el modelo. Un valor igual a 0 indica que la regresión no mejora la predicción basada exclusivamente en la media de la variable dependiente. Cuando el cálculo se realiza sobre datos de prueba, el indicador también puede adoptar valores negativos si las predicciones resultan menos precisas que aquella referencia básica (James et al., 2023).

La interpretación del coeficiente requiere una revisión conjunta con métricas expresadas en unidades del problema, como el error absoluto medio y el error cuadrático medio. Un valor elevado no demuestra causalidad, estabilidad temporal, ausencia de sesgos ni cumplimiento de los supuestos asociados con la regresión lineal. También puede aparecer una puntuación artificialmente alta cuando alguna variable predictora contiene información derivada de la respuesta o cuando los datos de prueba participan durante el ajuste del modelo (Chicco et al., 2021).

Figura 3

Representación conceptual del coeficiente de determinación



Elaboración propia con base en Chicco et al. (2021) y Montgomery et al. (2021). Los valores representados son ilustrativos.

6.4. Validación Cruzada (Cross-Validation)

La validación cruzada (cross-validation) es una técnica de evaluación estadística del rendimiento de los modelos de machine learning, que intenta corregir el principal inconveniente de la partición simple entre el conjunto de entrenamiento y el conjunto de prueba, que es la alta variabilidad del error estimado en función de cómo se haga la partición concreta sobre los datos. Para Arlot y Celisse (2010) la validación cruzada es un procedimiento de remuestreo que sistemáticamente divide el conjunto de datos del cual se dispone en cubatorias. Consiste en ajustar el modelo sobre una porción de los datos y evalúa su rendimiento predictivo para la porción restante, repitiendo el procedimiento rotativamente hasta que se ha entrenado y evaluado sobre todas las observaciones y, de esta manera, poder obtener una estimación robusta e imparcial del error de generalización del modelo.

La variante más utilizada es la validación cruzada (k-fold cross-validation), en la que una instancia de k pliegos (k-folds) con un tamaño aproximadamente igual divide el conjunto de datos. En cada iteración del procedimiento k−1 pliegos se utilizan para entrenar el modelo mientras que el siguiente se reserva para la evaluación. Este procedimiento se repite un total de k veces, de manera que cada pliegue actúa exactamente una vez como conjunto de evaluación. La estimación del error de generalización final se obtiene como la media aritmética de los k errores individuales calculados durante las distintas iteraciones, lo que permite una estimación más estable y menos sensible a una partición específica de los datos que la obtenida mediante una única división entre entrenamiento y prueba. Los valores de k más utilizados en la literatura son k = 5 y k = 10, lo que permite un buen compromiso entre el sesgo y la varianza del estimador del error (Sarker I. H., 2021).

6.5. Métricas de Error en Modelos de Regresión (MSE y MAE).

El error cuadrático medio (Mean Squared Error, MSE) cuantifica el promedio de los residuos elevados al cuadrado dentro del conjunto utilizado para evaluar el modelo. Esta operación concede mayor peso a las diferencias amplias y produce un resultado expresado en unidades cuadradas de la variable dependiente. Su formulación matemática corresponde a $MSE = (1/n)\Sigma(y_i - \hat{y}_i)^2$, donde n representa el número de observaciones evaluadas. (Malik, Khan, Abid, & Aslam, 2024).

El error absoluto medio (Mean Absolute Error, MAE) calcula el promedio de las magnitudes absolutas de las diferencias entre los valores observados y los valores estimados. La fórmula que representa el MAE es $MAE = (1/n)\Sigma|y_i - \hat{y}_i|$ y conserva las mismas unidades que la variable dependiente. La lectura conjunta del MSE, el MAE y el coeficiente de determinación ofrece una evaluación más completa del comportamiento predictivo, porque cada indicador representa una propiedad distinta del error (Chicco et al., 2021). (Olaitan, Adeniyi, & Onifade, 2024).

6.6. Preprocesamiento y Normalización de Datos

El preprocesamiento de datos constituye una etapa metodológica de carácter previo al entrenamiento de cualquier modelo de machine learning y representa, en la práctica, una de las fases que mayor impacto tiene sobre la calidad y la fiabilidad de los resultados obtenidos. La literatura especializada en ciencia de datos establece de forma consistente que la calidad del dato de entrada condiciona de manera directa y proporcional la calidad del modelo resultante, principio que en el ámbito anglosajón se sintetiza bajo la expresión garbage in, garbage out, y que adquiere especial relevancia cuando los datos de entrenamiento provienen de registros transaccionales comerciales que típicamente

presentan inconsistencias, valores faltantes, duplicados y escalas heterogéneas entre variables (Sarker I. H., 2021).

En el contexto del presente estudio, los datos históricos de ventas de alimento balanceado para camarón presentan características que hacen indispensable una etapa de preprocesamiento rigurosa antes de aplicar el algoritmo K-Means y los modelos de regresión por segmento. Entre las operaciones fundamentales de esta etapa se incluye el tratamiento de valores faltantes (missing values), que pueden originarse por ausencia de registros en períodos sin actividad comercial, errores de captura en el sistema de gestión o inconsistencias en la integración de fuentes de datos heterogéneas. Las estrategias más comunes para el tratamiento de valores faltantes comprenden la eliminación de las observaciones incompletas cuando su proporción es baja respecto al total del conjunto de datos, la imputación mediante la media o la mediana de la variable para datos con distribución simétrica o asimétrica respectivamente, y la imputación mediante modelos predictivos cuando la proporción de valores faltantes es significativa y su eliminación comprometería la representatividad de la muestra (Suyal & Sharma, A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning, 2024).

La normalización o estandarización de las variables numéricas constituye otra operación crítica del preprocesamiento, especialmente en el caso del algoritmo K-Means, cuya función de distancia euclidiana es altamente sensible a las diferencias de escala entre las variables incluidas en el análisis. Cuando las variables del conjunto de datos presentan rangos de magnitud muy distintos, por ejemplo una variable expresada en toneladas métricas y otra expresada en número de pedidos mensuales, la variable de mayor escala numérica dominará artificialmente el cálculo de las distancias y sesgará la formación de los clusters hacia la estructura de esa variable, independientemente de su relevancia

conceptual para la segmentación. Para corregir este problema se aplican dos técnicas principales: la normalización min-max, que transforma cada variable al intervalo [0, 1] mediante la expresión $x' = (x - x_{min}) / (x_{max} - x_{min})$, y la estandarización z-score, que centra cada variable en cero y la escala a desviación estándar unitaria mediante $x' = (x - \mu) / \sigma$, donde μ es la media y σ la desviación estándar de la variable. La estandarización z-score es generalmente preferida cuando las variables presentan distribuciones aproximadamente normales y cuando se prevé la existencia de valores atípicos que distorsionarían los límites de la normalización min-max (Sarker I. H., 2021).

7. Marco conceptual

El marco conceptual delimita los términos técnicos utilizados dentro del aprendizaje automático y establece un significado uniforme para cada procedimiento analítico. Esta delimitación evita interpretaciones contradictorias entre conceptos estadísticos, operaciones computacionales y medidas empleadas durante la evaluación de modelos. Las definiciones siguientes organizan los elementos necesarios para comprender la preparación, el entrenamiento y la evaluación de algoritmos supervisados y no supervisados.

7.1. Dataset (conjunto de datos)

Un dataset o conjunto de datos constituye una colección organizada de observaciones y variables que representa el fenómeno sometido a análisis. En una estructura tabular, cada fila suele corresponder a una observación individual, mientras cada columna representa una característica registrada para todas las observaciones. James et al. (2023) explican que los métodos estadísticos de aprendizaje utilizan estas estructuras para estimar relaciones, reconocer patrones y evaluar resultados sobre información no utilizada durante el ajuste.

El conjunto de entrenamiento, en el aprendizaje supervisado, puede representarse mediante una matriz X que contiene las variables predictoras y un vector y en el que se encuentra la variable objetivo. En el caso del aprendizaje no supervisado, el algoritmo trabaja fundamentalmente con la matriz de características porque no dispone de una respuesta conocida que oriente las asignaciones. La calidad del conjunto dependerá de su consistencia, representatividad, cobertura temporal y correspondencia con la finalidad analítica que se plantee.

7.2. Variable dependiente y variables independientes

La variable dependiente representa el resultado cuantitativo o categórico que un modelo supervisado procura estimar a partir de la información disponible. También recibe los nombres de variable de respuesta, variable objetivo o target dentro de la literatura relacionada con aprendizaje automático. En los problemas de regresión, esta variable adopta valores numéricos continuos y las predicciones conservan aquella naturaleza cuantitativa (James et al., 2023).

Las variables independientes aportan la información utilizada para estimar la respuesta y también se denominan predictores, características o features. El término variable predictora resulta más preciso cuando los datos proceden de registros observacionales, porque una asociación estadística no demuestra una relación causal. Cada predictor debe estar disponible durante el momento real de la predicción y no puede contener transformaciones derivadas de la respuesta que se pretende estimar.

7.3. Outlier (valor atípico)

Foorthuis (2021) define una anomalía como una ocurrencia que se aparta de los patrones generales presentes dentro de un conjunto de datos. Un valor atípico puede

originarse por errores de registro, mediciones excepcionales, cambios estructurales o acontecimientos legítimos de baja frecuencia. Por esta razón, una observación extrema no debe eliminarse automáticamente antes de establecer su origen y su significado dentro del dominio analizado.

Los outliers pueden modificar las medias, las varianzas, las distancias y los coeficientes estimados por diferentes métodos estadísticos. En K-Means, las observaciones extremas pueden desplazar los centroides y alterar la composición de los clústeres; en la regresión lineal, algunas observaciones pueden ejercer una influencia excesiva sobre la recta o el hiperplano estimado. El tratamiento debe especificar el criterio aplicado durante la revisión de los datos para distinguir los errores de captura de las operaciones reales poco frecuentes.

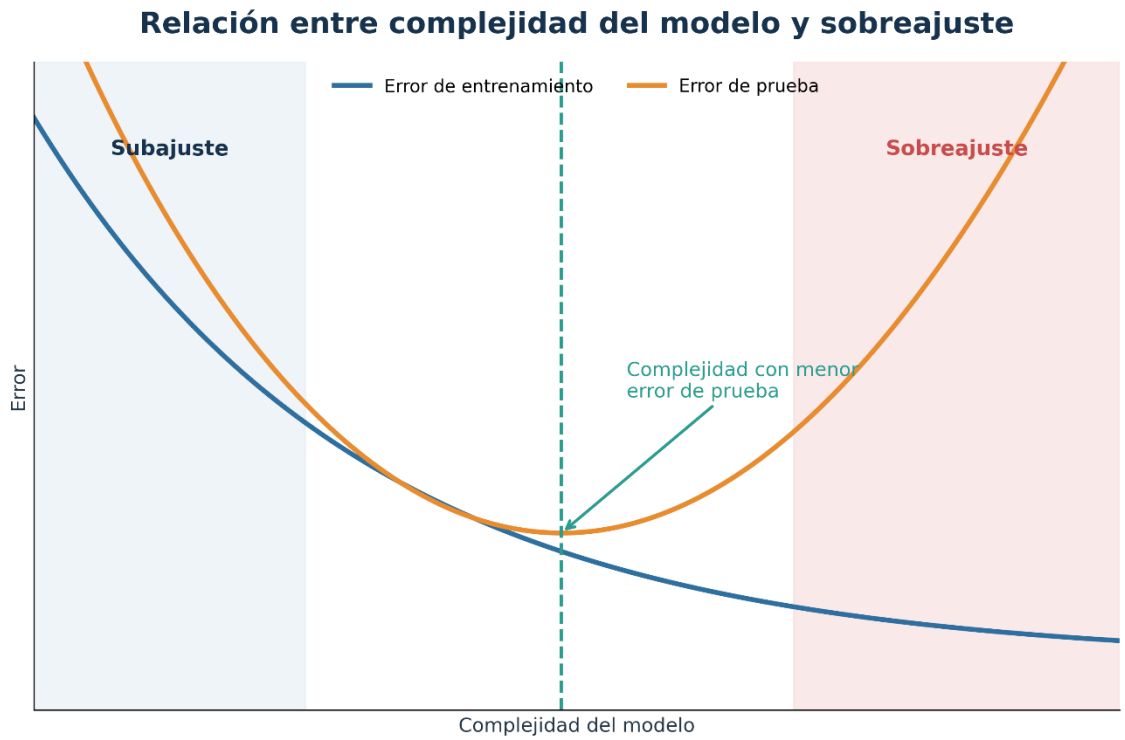
7.4. Overfitting (sobreajuste)

James et al. (2023) explican que el sobreajuste ocurre cuando un modelo reproduce con excesiva precisión las particularidades del conjunto de entrenamiento, pero pierde rendimiento ante observaciones nuevas. Esta situación aparece cuando la complejidad del modelo supera la cantidad o la calidad informativa disponible en los datos. El modelo aprende patrones reales junto con fluctuaciones aleatorias que no se mantienen fuera de la muestra utilizada durante el ajuste.

Una diferencia considerable entre el error de entrenamiento y el error de prueba constituye una señal frecuente de sobreajuste. La prevención requiere limitar la complejidad, seleccionar variables pertinentes, separar correctamente los conjuntos de datos y evaluar el rendimiento mediante procedimientos externos al ajuste. La validación cruzada puede complementar esta revisión cuando su aplicación aparece descrita y ejecutada de manera coherente dentro del procedimiento analítico.

Figura 4

Relación entre complejidad del modelo y sobreajuste



Elaboración propia con base en James et al. (2023).

7.5. Conjunto de entrenamiento y conjunto de prueba (train/test split)

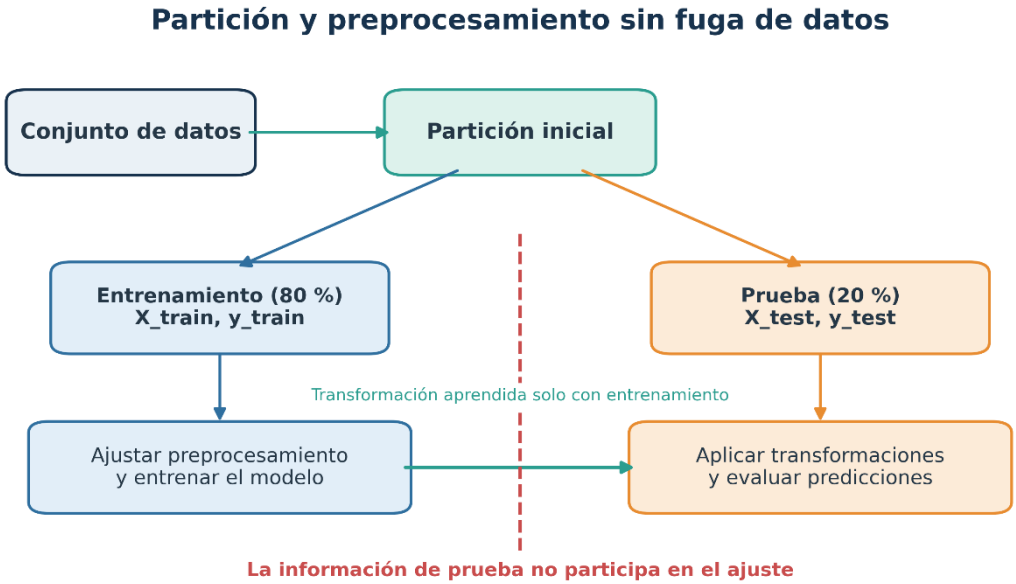
El conjunto de entrenamiento contiene las observaciones utilizadas para estimar parámetros, seleccionar representaciones y aprender los patrones que estructuran el modelo. El conjunto de prueba reserva observaciones diferentes para medir el comportamiento predictivo sobre datos que no participaron durante el aprendizaje. Esta separación proporciona una aproximación más realista a la capacidad de generalización que una evaluación realizada sobre los mismos registros empleados para ajustar el modelo (James et al., 2023).

La partición debe ejecutarse antes de calcular transformaciones cuyos parámetros dependan de la información disponible, como la imputación, la estandarización o la

selección de características. Tales operaciones se ajustan exclusivamente con los datos de entrenamiento y después se aplican al conjunto de prueba mediante los parámetros previamente estimados. Este orden evita que la información reservada influya sobre el modelo antes del cálculo final de las métricas (Scikit-learn developers, 2026a).

Figura 5

Partición y preprocesamiento sin fuga de datos.



Elaboración propia con base en James et al. (2023) y Scikit-learn developers (2026).

7.6. Preprocesamiento de datos

El preprocesamiento de datos comprende las operaciones destinadas a convertir registros originales en una estructura coherente para el análisis computacional. Estas operaciones abarcan la revisión de duplicados, el tratamiento de valores faltantes, la corrección de tipos, la codificación de categorías y el escalamiento de variables numéricas. La elección de cada transformación depende del algoritmo, la distribución de las variables y el significado sustantivo de los registros examinados (James et al., 2023).

La estandarización cobra una gran relevancia en los algoritmos basados en distancias, ya que las variables con escalas amplias pueden dominar el cálculo de similitud. Los parámetros de la transformación deben calcularse únicamente con el conjunto de entrenamiento para evitar que la información del conjunto de prueba influya en el proceso. Un procedimiento reproducible registra el orden de las operaciones y aplica siempre las mismas transformaciones durante el entrenamiento y la evaluación.

7.7. Generalización del modelo

La generalización representa la capacidad de un modelo para conservar un rendimiento adecuado frente a observaciones distintas de aquellas utilizadas durante el entrenamiento. Un modelo generaliza cuando aprende relaciones estables que permanecen fuera de la muestra original y evita depender de fluctuaciones particulares. Su evaluación requiere datos no utilizados durante el ajuste, métricas complementarias y procedimientos que respeten la independencia entre aprendizaje y comprobación (James et al., 2023).

8. MARCO LEGAL

Para procesar los registros comerciales mediante herramientas de aprendizaje automático es necesario distinguir la información empresarial de los datos que permiten identificar a una persona natural. La Ley Orgánica de Protección de Datos Personales define como dato personal toda aquella información que identifica o permite identificar, directa o indirectamente, a una persona natural. Su ámbito incluye las operaciones automatizadas y no automatizadas que se efectúan sobre esta clase de información. Por esta razón, los códigos de clientes, identificadores de vendedores y demás referencias asociables con personas naturales deben recibir protección cuando permitan establecer su identidad, mientras que los valores comerciales agregados mantienen una naturaleza

diferente cuando no permiten esa identificación (Ley Orgánica de Protección de Datos Personales [LOPDP], 2021, arts. 2 y 4).

La LOPDP (2021, art. 10) regula los principios de juridicidad, lealtad, transparencia, finalidad, pertinencia y minimización, proporcionalidad, confidencialidad, calidad y exactitud, conservación, seguridad y responsabilidad proactiva para el tratamiento de datos personales. Estos principios imponen que la información utilizada esté dirigida a una finalidad determinada y que cada variable guarde una relación razonable con el propósito analítico establecido. En una estimación de ventas, los identificadores que no sirven para explicar el comportamiento comercial deben mantenerse fuera de las matrices analíticas cuando no se requiera conservarlos. La misma disposición exige medidas de seguridad apropiadas a la naturaleza de los datos, al contexto del tratamiento y a los riesgos que puedan afectar a sus titulares.

El Reglamento General de la Ley Orgánica de Protección de Datos Personales desarrolla estas obligaciones mediante criterios orientados al control de riesgos y la protección de la información desde el diseño. Conforme a los artículos 58, 59 y 60, el responsable debe implementar las medidas técnicas, jurídicas, administrativas y organizativas adecuadas, así como limitar el tratamiento a los datos necesarios para cada finalidad específica. La protección desde el diseño implica anticipar los riesgos antes del tratamiento, en tanto que la protección por defecto limita el acceso y disminuye la exposición de datos personales durante las diversas etapas del procesamiento (Presidencia de la República del Ecuador, 2023). Estas disposiciones cobran particular importancia cuando las bases comerciales atraviesan procesos de limpieza, transformación, codificación, almacenamiento y análisis automatizado.

La reducción de identificadores es importante cuando el análisis estadístico no requiere saber quiénes son los clientes, vendedores u otras personas vinculadas a las operaciones comerciales. La Superintendencia de Protección de Datos Personales (2025) define la anonimización como una medida técnica orientada a impedir la identificación o reidentificación de una persona natural sin esfuerzos desproporcionados. Esta misma regulación distingue este procedimiento de la seudonimización, en el que los datos personales son reemplazados por códigos que aún pueden ser vinculados al titular mediante información adicional almacenada por separado. Por ello, el hecho de sustituir un nombre por un código no hace que automáticamente el registro pase a ser información anónima cuando exista un mecanismo para reconstruir la identidad original.

La Resolución N.º SPDP-SPD-2025-0030-R establece lineamientos específicos sobre la anonimización, seudonimización, bloqueo y eliminación dentro del ciclo de vida de los datos personales. Esta regulación permite establecer controles diferenciados en función del grado de identificación que conserve la base una vez preparada para su tratamiento estadístico. Cuando no son necesarios para agrupar o estimar valores comerciales, dejar fuera los identificadores personales reduce los riesgos de acceso no autorizado y limita la capacidad de vincular las predicciones a individuos específicos. Por tanto, la conservación de variables debe regirse por su utilidad analítica y por el principio de minimización establecido por la legislación ecuatoriana (Superintendencia de Protección de Datos Personales [SPDP], 2025).

Desde 2026, además, se encuentra regulado el uso de sistemas de inteligencia artificial que traten datos personales. La Resolución N.º SPDP-SPD-2026-0009-R dispone que los responsables y encargados que desarrollen, entrenen, implementen o utilicen estos sistemas deben respetar los principios, derechos y obligaciones establecidos por la LOPDP y su Reglamento General. La norma exige informar las finalidades del

tratamiento automatizado, gestionar sus riesgos, aplicar las medidas de seguridad en función del volumen y las categorías de datos, registrar las actividades correspondientes y realizar controles sobre el funcionamiento del sistema (SPDP, 2026, arts. 1, 5 y 7). Estas obligaciones son aplicables cuando el procesamiento automatizado utilice datos personales de titulares ecuatorianos y no cuando el sistema opere exclusivamente con información que queda fuera del ámbito material de la LOPDP.

La regulación de 2026 también salvaguarda el derecho de las personas frente a decisiones basadas en procesos automatizados cuando estas produzcan efectos jurídicos o incidan en sus derechos y libertades. El artículo 4 obliga a respetar los derechos a la información, oposición y protección frente a determinadas valoraciones automatizadas cuando el sistema trate datos personales. En una aplicación orientada a la estimación de ventas, los resultados obtenidos mediante K-Means y modelos de regresión deben centrarse en patrones transaccionales y valores de venta, sin extenderse a valoraciones personales que no se correspondan con la finalidad económica definida. Esta limitación mantiene el procesamiento dentro del propósito para el cual se seleccionaron las variables y reduce riesgos relacionados con usos posteriores de la información (SPDP, 2026).

En este sentido, el marco normativo ecuatoriano define una relación directa entre la finalidad del análisis, la selección de las variables y las medidas para proteger los registros. Los datos necesarios para estimar ventas pueden conservarse cuando respondan al propósito comercial establecido, mientras que los identificadores sin utilidad analítica requieren su exclusión, anonimización o un tratamiento restringido según corresponda. Durante la preparación de la información, el procesamiento mediante algoritmos y la presentación de los resultados, se deben mantener la seguridad, la minimización, la confidencialidad y el control del riesgo. De esta manera, el empleo de K-Means y regresión lineal múltiple puede orientarse al análisis de transacciones y a la estimación de

Total USD sin extenderse al tratamiento de datos personales ajenos a la finalidad comercial.

9. METODOLOGÍA

El presente capítulo expone la ruta seguida para preparar los registros comerciales, formar grupos mediante K-Means y estimar el valor de las ventas mediante regresión lineal múltiple. La organización del capítulo respeta el orden del cuaderno desarrollado en Google Colab, desde la revisión inicial hasta la evaluación del modelo predictivo. El procedimiento también incorpora controles estadísticos para evitar información repetida, variables identificadoras dentro de los modelos y datos derivados de la respuesta durante el agrupamiento.

9.1. Enfoque, tipo, alcance y diseño de la investigación

Creswell y Creswell (2023) señalan que el enfoque cuantitativo estudia relaciones mediante datos numéricos y procedimientos definidos antes del análisis. Esta investigación adoptó dicho enfoque porque examinó registros históricos expresados en cantidades, precios, descuentos, fechas y valores monetarios. La elección permitió describir patrones comerciales y estimar Total USD con técnicas estadísticas aplicadas sobre información empresarial.

El estudio tuvo carácter aplicado porque atendió una necesidad concreta de planificación comercial dentro de una empresa distribuidora de alimento balanceado para camarón. Sarker (2021) explica que el aprendizaje automático obtiene patrones desde datos disponibles para apoyar decisiones relacionadas con problemas reales. En este caso, el análisis buscó ofrecer una base cuantitativa para revisar ventas, demanda, inventarios y decisiones comerciales.

El alcance fue descriptivo y predictivo porque cada fase respondió a una finalidad distinta dentro del mismo estudio. La fase descriptiva revisó la estructura de los datos, los valores ausentes, los valores atípicos y los grupos formados por K-Means. La fase predictiva estimó Total USD mediante regresión lineal múltiple y comparó los valores observados con las predicciones obtenidas.

El periodo disponible comprendió desde el 2 de enero de 2025 hasta el 20 de mayo de 2026. Esta amplitud temporal permitió representar cambios mediante año, mes, semana, día y día de la semana.

9.2. Fuente de datos, población y unidad de análisis

La fuente correspondió a una base interna de una empresa dedicada a comercializar alimento balanceado para camarón. El archivo original reunió 7.216 registros y 38 variables relacionadas con clientes, productos, fechas, cantidades, precios, descuentos, vendedores y operaciones comerciales. La base presentó una estructura tabular donde cada fila representó una línea de transacción registrada por la empresa.

La población estuvo formada por todos los registros disponibles dentro del periodo cubierto por el archivo empresarial. No se aplicó muestreo porque el análisis tomó la totalidad de las filas entregadas para el estudio. Esta decisión conservó operaciones frecuentes y poco frecuentes que podían aportar información durante la formación de los grupos.

Cada registro transaccional asociado con una venta, devolución, rebaja, nota de crédito u otro movimiento comercial constituyó la unidad de análisis.NUn único cliente podía quedar reflejado en el histórico en varias fechas, productos y cantidades, por lo que sus operaciones podían estar contenidas en filas distintas y, en consecuencia, cada fila no

representó una persona diferente. Dicha precisión permitió interpretar los clústeres como grupos de transacciones con rasgos semejantes dentro del histórico analizado.

La compañía fue identificada con el nombre de Empresa ABC con el propósito de proteger su información de tipo comercial durante la elaboración del trabajo. Los códigos, los nombres internos, las rutas, los clientes y los vendedores tuvieron carácter reservado durante todo el proceso académico. Los resultados fueron presentados de forma consolidada para evitar la identificación directa de personas u operaciones concretas.

9.3. Entorno computacional y recursos empleados

El procesamiento fue llevado a cabo en Google Colab, mediante un cuaderno con extensión .ipynb y código escrito en Python. Tal entorno sirvió para ejecutar cada una de las fases de forma ordenada, guardar las salidas y poder revisar los cambios aplicados sobre la matriz. La secuencia pasó por la carga, inspección, limpieza, transformación, agrupamiento, preparación predictiva, regresión y evaluación estadística.

Las librerías pandas y NumPy se utilizaron para la tarea de lectura, revisión, ordenación y transformación de los registros comerciales existentes. Las librerías Matplotlib y seaborn apoyaron la presentación del método del codo, del coeficiente de silueta, de las predicciones y de los residuos. La librería scikit-learn aportó StandardScaler, KMeans, train_test_split, LinearRegression y las métricas definidas para evaluar el modelo.

La reproducción del análisis se apoyó en parámetros fijos dentro de los procesos con selección aleatoria. El valor random_state se estableció en 42 durante la partición de los datos y la formación de los clústeres. El modelo K-Means también usó diez inicializaciones mediante n_init igual a 10 para comparar distintas posiciones iniciales.

9.3.1. Funciones, clases y métodos de Python empleados

Se realizó el procesamiento de los datos y la implementación de los modelos mediante las funciones, clases y métodos que ofrecen Python y sus librerías más importantes para el aprendizaje automático y análisis de datos. Pandas permitió hacer la preparación e inspección inicial de la base a través de DataFrame.info(), método que permite examinar la estructura, las clases de datos y el número de valores disponibles por variable; DataFrame.describe(), método que permite obtener estadísticas descriptivas de las variables numéricas; y DataFrame.isnull(), método que permite detectar los valores faltantes en los registros. Las operaciones permitieron examinar los atributos de la matriz, antes de aplicar las técnicas de modelado y transformación.

Para la preparación de las variables a usar por K-Means se utilizó la clase StandardScaler de sklearn.preprocessing, la cual tiene como función transformar las características numéricas a una escala comparable mediante su media y desviación estándar. Luego se utilizó la clase KMeans de sklearn.cluster para realizar el proceso de agrupamiento de las transacciones. El modelo se configuró con dos clústeres, diez inicializaciones y un random_state igual a 42, tal como se especifica en el script. La matriz usada para esta fase tomó en cuenta las variables comerciales elegidas, entre ellas Total USD, de conformidad con el procedimiento metodológico establecido para el estudio.

La configuración, en la que se dividieron los registros en un conjunto de entrenamiento correspondiente al 80 % y en un conjunto de prueba que abarcó el 20 %, se realizó a través de la función train_test_split de sklearn.model_selection con un random_state igual a 42. Posteriormente, se aplicó la clase LinearRegression de sklearn.linear_model para el ajuste de una regresión lineal múltiple general sobre las observaciones correspondientes a los dos clústeres. La variable Total USD actuó como

respuesta del modelo, mientras que se incluyó entre las variables explicativas empleadas para generar las estimaciones la etiqueta Clúster obtenida mediante K-Means.

9.4. Organización y descripción de las variables

La organización inicial separó las variables según su naturaleza, su significado comercial y su función dentro del procedimiento. Esta clasificación permitió distinguir identificadores, categorías, medidas cuantitativas, datos temporales y campos económicos. La revisión también facilitó decidir cuáles variables debían conservarse, transformarse, codificarse o retirarse antes del modelado.

Tabla 1

Diccionario de las variables contenidas en la base original

N.°	Variable	Tipo inicial	Descripción y tratamiento metodológico
1	División	Categórica	Unidad operativa registrada; se retiró durante el paso ocho por su función administrativa.
2	Código de planta	Categórica	Identificador de la planta; se retiró porque no representó una medida comercial.
3	Código de almacén	Categórica	Identificador del almacén; se retiró antes de la codificación del modelo.
4	Categoría Cliente	Categórica	Categoría comercial del cliente; se conservó y recibió codificación binaria.
5	Código de cliente	Numérica identificadora	Identificador del cliente; se convirtió a entero y quedó fuera de K-Means y regresión.
6	Clase de cliente	Categórica	Clase comercial asignada al cliente; se conservó mediante codificación binaria.

N.°	Variable	Tipo inicial	Descripción y tratamiento metodológico
7	Nombre de grupo	Categórica	Grupo comercial relacionado con el cliente; se conservó para representar diferencias empresariales.
8	Nombre completo de cliente	Categórica identificadora	Identificador nominal; se mantuvo únicamente para trazabilidad y quedó fuera de los modelos.
9	City	Categórica	Ciudad relacionada con la operación; se conservó como variable geográfica.
10	Vendedor Local Asignado	Categórica identificadora	Nombre interno del vendedor asignado; se retiró durante el paso ocho.
11	Código de vendedor - Transacción	Numérica identificadora	Código del vendedor; se retiró porque presentó un único valor dentro de los registros.
12	Vendedor Transacción	Categórica identificadora	Nombre del vendedor vinculado con la operación; se retiró durante el paso ocho.
13	Mes	Numérica temporal	Mes almacenado en la base; se comparó con la fecha y se conservó una sola representación mensual.
14	Semana	Numérica temporal	Semana comercial registrada; se conservó como referencia temporal complementaria.
15	Número de factura	Categórica identificadora	Identificador único de factura; se retiró durante el paso ocho.
16	Fecha de factura	Fecha	Fecha de cada operación; permitió obtener año, mes, día y día semanal.
17	Número de pedido	Numérica identificadora	Identificador de pedido; se excluyó del modelado por carecer de significado métrico.
18	Número de pedido del cliente	Categórica identificadora	Referencia externa del pedido; se retiró por sus 2.584 valores ausentes.

N.°	Variable	Tipo inicial	Descripción y tratamiento metodológico
19	Área de negocio	Categórica	Área comercial relacionada con la venta; se conservó mediante codificación binaria.
20	Código de artículo	Categórica	Código del producto; se conservó para diferenciar los artículos vendidos.
21	Especie	Categórica	Especie acuícola relacionada con el alimento; se conservó dentro de la matriz.
22	Línea de Producto	Categórica	Línea comercial del producto; se conservó para representar familias de alimento.
23	SG3 Species L4	Numérica vacía	Campo jerárquico sin datos; se retiró porque sus 7.216 observaciones estaban ausentes.
24	Presentación de Producto	Categórica	Presentación comercial del producto; se conservó mediante codificación binaria.
25	Nombre de artículo	Categórica	Denominación del artículo; se conservó para reconocer diferencias entre productos.
26	Packing Sales	Categórica	Forma de empaque registrada en la venta; se conservó mediante codificación binaria.
27	Sacos	Numérica continua	Cantidad de sacos relacionada con la transacción; se conservó como medida de volumen.
28	Kilos	Numérica continua	Peso total expresado en kilogramos; se conservó como medida principal de volumen.
29	TM	Numérica continua	Peso expresado en toneladas métricas; se retiró del modelo por su relación exacta con Kilos.

N.°	Variable	Tipo inicial	Descripción y tratamiento metodológico
30	Precio Unit.	Numérica continua	Precio unitario del producto; se conservó como variable económica explicativa.
31	Dcto. por Saco	Numérica continua	Descuento monetario aplicado por saco; se conservó después de revisar su consistencia.
32	Total USD	Numérica continua	Valor monetario de la transacción; actuó como variable dependiente de la regresión.
33	Dcto. Calculado	Numérica continua	Valor total del descuento calculado; se conservó como variable económica.
34	% Dcto	Numérica continua	Porcentaje de descuento aplicado; se conservó después de revisar su escala.
35	Ruta	Categórica identificadora	Código de la ruta comercial; se retiró durante el paso ocho.
36	Ruta Descripción	Categórica identificadora	Nombre de la ruta; se retiró por repetir información logística del código.
37	Tipo de pedido - Código	Categórica	Tipo de operación, como venta, devolución o rebaja; se conservó mediante codificación binaria.
38	Tamaño Partícula	Categórica	Tamaño físico del alimento; se conservó para representar diferencias del producto.

*Nota***.** La descripción se elaboró a partir de los nombres, tipos y salidas registradas dentro del cuaderno de Google Colab.

Descriptivo de variables

Tabla 2

Distribución porcentual de las variables seleccionadas de la base de datos

Grupo	Variable	Categoría o valor	Porcentaje
Cliente y localización	Categoría Cliente	Farmers - Aqua	100 %
Cliente y localización	Código de cliente	10035	9,77 %
		10036	7,60 %
		10029	7,58 %
		Otros	75,05 %
Cliente y localización	Clase de cliente	Local	100 %
Cliente y localización	Nombre de grupo	Grupo P	19,67 %
		Grupo I	16,55 %
		Grupo Z	12,21 %
		Otros	51,57 %
Cliente y localización	City	Machala	50,51 %
		Pasaje	15,10 %
		El Guabo	11,06 %
		Otros	23,33 %
Operación comercial	Código de vendedor - Transacción	495	100 %
Tiempo	Mes	4	13,24 %
		1	11,63 %
		2	11,30 %
		Otros	63,83 %
Tiempo	Semana	17	3,42 %
		18	3,38 %
		13	3,05 %
		Otros	90,15 %
Producto y presentación	Área de negocio	AQUA	100 %

Grupo	Variable	Categoría o valor	Porcentaje
Producto y presentación	Código de artículo	14385805	27,64 %
		17737185	8,65 %
		CRE-001	7,83 %
		Otros	55,88 %
Producto y presentación	Especie	Camarón	100 %
Producto y presentación	Línea de Producto	Estándar	53,81 %
		Premium	19,44 %
		Salud	9,55 %
		Iniciador	9,09 %
		Ecoline	8,11 %
Producto y presentación	Presentación de Producto	Extruido	97,25 %
		Palletizado	2,22 %
		No aplica	0,52 %
Producto y presentación	Packing Sales	25KG/BAG	99,47 %
		No usado	0,53 %
Volumen y precio	TM	2500	7,03 %
		5000	6,38 %
		7500	6,33 %
		Otros	80,26 %
Volumen y precio	Precio Unit.	25,05	11,47 %
		30,37	4,88 %
		26,05	3,37 %
		Otros	80,28 %
Operación comercial	Tipo de pedido - Código	Ventas normales	91,06 %
		Devoluciones	4,34 %

Grupo	Variable	Categoría o valor	Porcentaje
		Diferencia de precio	2,34 %
		Otros	2,26 %
Producto y presentación	Tamaño Partícula	#5	62,82 %
		#4	9,73 %
		#2	8,79 %
		#1	8,57 %
		#0	7,79 %
		#3	2,26 %
		#6	0,04 %

*Nota***.** Los porcentajes corresponden a la distribución observada de las categorías o valores seleccionados dentro de los registros de la base de datos.

Las variables se agruparon en cinco conjuntos para ordenar el procesamiento posterior. El primer conjunto reunió datos del cliente y su localización; el segundo reunió información del producto y su presentación. Los tres conjuntos restantes reunieron tiempo, volumen, precio, descuento y tipo de operación comercial.

Se definió como variable dependiente el Total USD ya que representó el valor monetario que posteriormente debía estimar la regresión. También se preservó esta variable dentro de la matriz empleada para ejecutar K-Means, tal como se configuró en el procedimiento utilizado. La etiqueta Clúster surgió luego del proceso de agrupamiento y posteriormente se utilizó como una variable explicativa adicional dentro del modelo general de regresión lineal múltiple.

9.5. Procesamiento y preparación de los datos

9.5.1. Inspección inicial de la matriz

La primera fase examinó la forma general de la matriz antes de aplicar cualquier transformación. El comando info permitió revisar dimensiones, nombres, tipos de datos y número de valores disponibles por variable. El comando describe mostró medidas de posición y dispersión para reconocer escalas diferentes, valores negativos y observaciones extremas.

La inspección confirmó 7.216 filas y 38 columnas dentro del archivo original. La matriz presentó una fecha, catorce variables numéricas y veintitrés variables categóricas antes de las primeras correcciones. Esta revisión proporcionó la base necesaria para ordenar los pasos posteriores sin alterar campos de forma innecesaria.

9.5.2. Limpieza general de los datos

La limpieza general tuvo un desarrollo breve porque su propósito fue preparar la información para el agrupamiento y la regresión. El proceso corrigió tipos de datos, revisó ausencias, retiró campos vacíos y verificó la consistencia de los registros. Esta etapa evitó que problemas básicos de estructura afectaran las distancias y los coeficientes calculados después.

Sarker (2021) explica que la calidad de los datos condiciona el resultado de los algoritmos aplicados sobre ellos. Por esta razón, el procedimiento conservó un orden fijo entre revisión, corrección, codificación y escalado. La matriz pasó a la etapa de modelado únicamente después de completar las verificaciones previstas.

9.5.3. Identificación y tratamiento de valores faltantes

Peng et al. (2022) señalan que los valores ausentes pueden afectar la calidad de los datos y producir sesgos cuando reciben un tratamiento sin justificación. El script contó los valores nulos de cada columna mediante isnull y ordenó los resultados desde la mayor ausencia. Esta revisión permitió diferenciar columnas completamente vacías, campos con alta ausencia y variables con pocos datos faltantes.

SG3 Species L4 presentó 7.216 valores ausentes, por lo cual se retiró completamente de la base. Número de pedido del cliente presentó 2.584 ausencias y también se retiró por su función identificadora y su baja disponibilidad. Esta decisión redujo columnas sin información suficiente antes de las transformaciones numéricas.

Los registros con ausencia simultánea en el código del cliente, la fecha y varios campos transaccionales recibieron una revisión conjunta antes del modelado. Cuando una fila no permitió reconocer una operación completa, el registro fue retirado para evitar la creación artificial de una transacción. Los identificadores no recibieron medias ni medianas porque sus números carecían de significado cuantitativo.

Las variables numéricas con ausencias aisladas recibieron la mediana de su columna después de confirmar la validez del registro restante. Las variables categóricas incompletas recibieron la moda, mientras las fechas ausentes se revisaron frente a los demás datos de la operación. Después de estos controles, la matriz quedó completa para las etapas de selección y codificación.

9.5.4. Conversión de tipos de datos

La conversión de tipos corrigió columnas cuyo formato inicial no representaba su función real dentro del archivo. Código de cliente pasó desde un formato decimal hacia

un formato entero porque actuaba como identificador interno. Fecha de factura conservó un formato temporal para permitir la extracción posterior de componentes cronológicos.

Los códigos numéricos no fueron tratados como medidas continuas dentro de K-Means o regresión. Un código mayor no representa una venta mayor, una frecuencia superior ni un cliente más importante. Esta separación evitó que números creados para identificar registros aportaran distancias artificiales dentro de los modelos.

9.5.5. Detección y tratamiento de valores atípicos

La detección de valores atípicos empleó el rango intercuartílico sobre variables cuantitativas con significado comercial. El procedimiento calculó los cuartiles primero y tercero, además de la diferencia existente entre ambos valores. La relación matemática usada para obtener el rango intercuartílico fue la siguiente:

$$RIQ = Q_3 - Q_1$$

Los límites inferior y superior se calcularon con una distancia equivalente a una vez y media el rango intercuartílico. Las expresiones permitieron señalar observaciones alejadas del centro de cada distribución cuantitativa. Los límites aplicados durante la revisión se expresaron mediante las siguientes ecuaciones:

$$L_i = Q_1 - 1{,}5(RIQ)$$

$$L_s = Q_3 + 1{,}5(RIQ)$$

Se procedió a aplicar un análisis mediante el rango intercuartílico sobre Sacos, Kilos, Precio Unitario, Dcto. por Saco, Dcto. Calculado, % Dcto y Total USD. El Código de cliente, Código de vendedor, Número de pedido, mes, semana y demás identificadores

quedaron fuera de este tratamiento, limitación impuesta para no modificar códigos válidos que carecen de significado cuantitativo continuo.

Los valores mencionados fueron sometidos a una revisión en función del tipo de pedido y de la lógica comercial de la operación. Se mantuvieron las cantidades negativas relacionadas con devoluciones, descuentos o notas de crédito, pues correspondían a movimientos comerciales válidos. El ajuste a los límites se reservó para valores extremos sin respaldo comercial o para errores confirmados durante la revisión

9.5.6. Eliminación de variables irrelevantes

El paso ocho del script retiró variables que cumplían una función administrativa, identificadora o repetida dentro del archivo comercial. Las columnas retiradas fueron División, Código de planta, Código de almacén, Número de factura, Vendedor Local Asignado, Vendedor Transacción, Ruta y Ruta Descripción. La presente selección delimitó variables que podían ampliar la matriz codificada, pero que no aportaban una medida directa sobre la transacción.

El número de factura se eliminó puesto que identificaba determinadas operaciones y contenía una elevada cantidad de categorías. Las variables de vendedor y ruta incluían referencias internas que no participaban en la predicción planteada. Los códigos de planta, almacén y división también se eliminaron porque describían la estructura administrativa de la compañía.

El control metodológico retiró además Código de vendedor - Transacción porque todos los registros mostraron el valor 495. Número de pedido, Código de cliente y Nombre completo de cliente quedaron fuera de los modelos por su función identificadora.

TM también se retiró de las matrices porque representaba Kilos dividido para mil y generaba información lineal repetida.

9.5.7. Codificación de variables categóricas

James et al. (2023) explican que los modelos numéricos requieren una representación adecuada para las variables categóricas. El procedimiento aplicó codificación One-Hot sobre las categorías seleccionadas después de la eliminación de identificadores. Cada categoría generó una columna binaria que indicó presencia o ausencia dentro de cada registro.

La codificación puede representarse mediante una variable indicadora que toma uno cuando la observación pertenece a una categoría específica. El valor cero indica que la categoría no corresponde al registro evaluado dentro de la matriz. La expresión general empleada para esa transformación fue la siguiente:

$$d_{ij} = \begin{cases} 1, cuando\ la\ observación\ i\ pertenece\ a\ la\ categoría\ j, \\ 0, cuando\ la\ observación\ i\ no\ pertenece\ a\ la\ categoría\ j. \end{cases}$$

El parámetro drop_first eliminó una categoría de referencia dentro de cada conjunto categórico codificado. Esta decisión redujo columnas repetidas y disminuyó relaciones exactas entre variables indicadoras dentro de la regresión. La matriz resultante quedó formada por variables cuantitativas y columnas binarias aptas para el procesamiento posterior.

9.5.8. Estandarización de las características

Perucha Jurjo (2022) explica que K-Means forma grupos a partir de distancias entre observaciones y representantes. Las diferencias de escala pueden provocar que una variable con números grandes domine el resultado del agrupamiento. Por esta razón, las

características seleccionadas recibieron una estandarización antes de ejecutar el algoritmo.

StandardScaler transformó cada valor mediante la media y la desviación estándar de su variable. Esta operación produjo columnas con media cercana a cero y desviación estándar cercana a uno. La fórmula aplicada para cada valor de la matriz fue la siguiente:

$$z_{ij} = \frac{(x_{ij} - \bar{x}_j)}{(s_j)}$$

En esta expresión, x_{ij} representa el valor original de la observación, mientras $\bar{x}_j$ representa la media de la variable. El término s_j corresponde a la desviación estándar calculada sobre la columna respectiva. La transformación permitió comparar variables que originalmente estaban expresadas mediante escalas comerciales bastante diferentes.

Total USD participó en la matriz destinada al proceso de agrupamiento mediante K-Means, y se le dio el tratamiento correspondiente para mantener una escala comparable con el resto de las variables cuantitativas empleadas. Los identificadores y la fecha original no fueron parte directa del cálculo de las distancias entre registros, puesto que no constituían características métricas apropiadas para realizar el agrupamiento. Esta preparación permitió crear la matriz que posteriormente fue empleada por el algoritmo.

9.6. Aplicación del algoritmo K-Means

9.6.1. Matriz usada para el agrupamiento

La matriz que se empleó para K-Means agrupó las variables comerciales escogidas tras los procesos de limpieza, transformación, codificación y estandarización. Entre las características consideradas durante el agrupamiento se mantuvo Total USD, junto con las demás variables cuantitativas y categóricas transformadas que aportaban

información sobre las transacciones. Los identificadores internos, nombres personales y variables repetidas linealmente quedaron fuera de esta fase, por no constituir características adecuadas para el cálculo de las distancias.

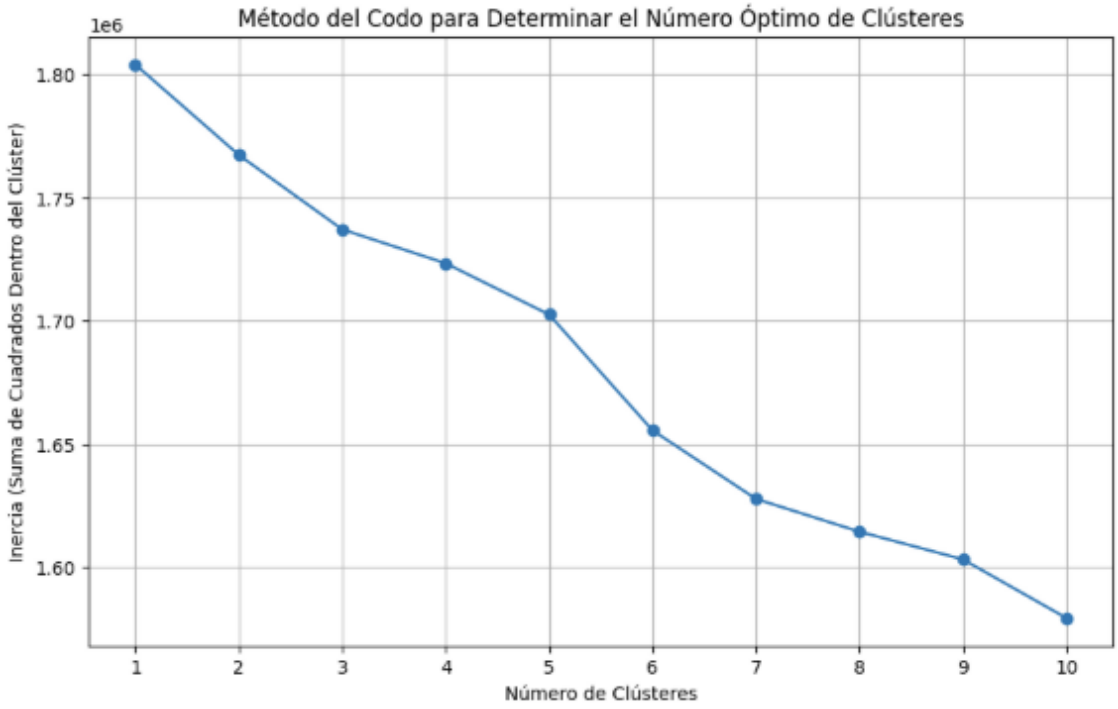
Esta configuración permitió que los clústeres se formaran tomando conjuntamente en cuenta las características comerciales y monetarias presentes en los registros. La etiqueta generada por K-Means reflejaba a qué uno de los grupos identificados pertenecía cada transacción y posteriormente se incluyó como una variable explicativa en el modelo general de regresión lineal múltiple.

9.6.2. Determinación del número de clústeres

El número de clústeres fue revisado mediante el método del codo y el coeficiente de silueta. El método del codo calculó la inercia para valores de k comprendidos entre uno y diez. La curva permitió observar el punto donde una división adicional produjo una reducción menor de la dispersión interna. La inercia correspondió a la suma de distancias cuadráticas entre cada observación y el centroide asignado. Una reducción grande indicó que el nuevo grupo explicaba una parte relevante de la variación interna. Una reducción pequeña mostró que el aumento de k aportaba poca mejora sobre la estructura previa.

Figura 6

Método del codo para determinar el número óptimo de clústeres



Nota. La inercia representa la suma de las distancias cuadráticas entre cada observación y el centroide del grupo asignado. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

El comportamiento de la curva permitió verificar la reducción de la inercia a medida que se incrementaba el número de clústeres. Si bien la disminución continuó durante los valores evaluados, la gráfica no mostró un punto de inflexión completamente definido. Por este motivo, la selección del número de grupos se complementó con el coeficiente de silueta.

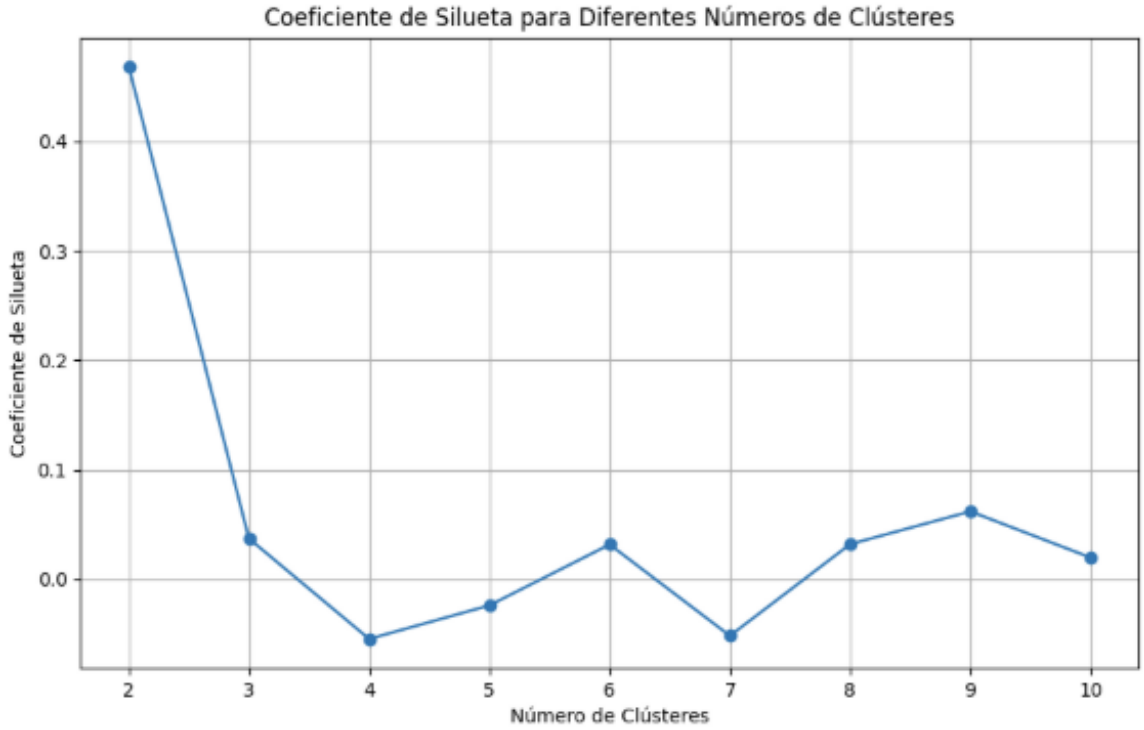
El coeficiente de silueta comparó la cercanía de cada registro con su propio grupo y con el grupo vecino. Los valores cercanos a uno representaron una separación mayor, mientras los valores cercanos a cero señalaron solapamiento. La expresión aplicada para cada observación fue la siguiente:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

En esta expresión, a(i) representa la distancia media entre la observación y los miembros de su propio grupo. El término b(i) representa la menor distancia media frente a los miembros de otro grupo. El valor definitivo de k se seleccionó después de comparar ambos criterios sobre la matriz corregida y estandarizada, que incorporó las variables definidas para el agrupamiento, entre ellas Total USD.

Figura 7

Coeficiente de silueta para diferentes números de clústeres



Nota. El coeficiente de silueta compara la cohesión interna de cada grupo con su separación respecto de los demás grupos. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

El coeficiente de silueta alcanzó su mejor valor cuando el número de clusters fue igual a dos. Los valores obtenidos para el resto de configuraciones fueron bastante inferiores, en algunos casos negativos. Este resultado respaldó la elección de dos grupos para ejecutar el modelo K-Means sobre la matriz preparada.

9.6.3. Ajuste del modelo y asignación de los grupos

El modelo final se configuró con el número de clústeres seleccionado, diez inicializaciones y una semilla aleatoria igual a 42. Las distintas inicializaciones permitieron comparar varias posiciones de partida antes de conservar la solución con menor inercia. El algoritmo recibió la matriz estandarizada y asignó una etiqueta a cada transacción disponible.

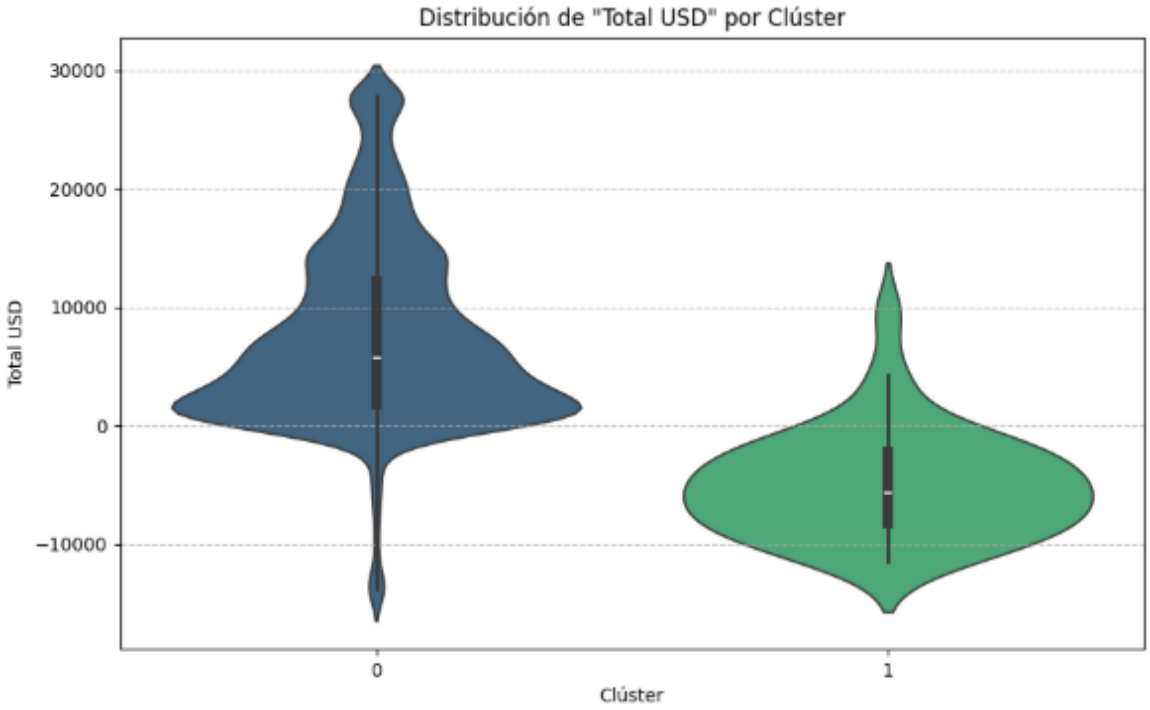
La primera operación consistía en comparar cada registro con los centroides definidos para la iteración en cuestión. Cada una de las observaciones recibía el grupo cuyo centroide tenía la menor distancia euclidiana cuadrática. La segunda operación consistía en sustituir cada centroide por la media de las observaciones que habían sido asignadas a ese grupo. El proceso se repetía mediante las fases de asignación y actualización hasta alcanzar una partición estable o hasta que se llegase al criterio de parada del algoritmo. Posteriormente, se añadía la columna Clúster al conjunto original para identificar a qué grupo pertenecía cada transacción. Finalmente, esta variable se utilizó como predictor categórico dentro de la regresión lineal múltiple general para apoyar la estimación de Total USD.

9.6.4. Caracterización de los clústeres

La caracterización comparó medias cuantitativas y proporciones de categorías entre los grupos obtenidos. Las variables con mayores diferencias permitieron describir el volumen, precio, descuento, producto y tipo de pedido asociado con cada conjunto. Este examen ofreció una interpretación comercial de las etiquetas numéricas asignadas por K-Means.

Figura 8

Distribución de Total USD según el clúster asignado



Nota. El ancho de cada figura representa la concentración de observaciones dentro de los diferentes valores de Total USD. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

La distribución de Total USD, mostró diferencias entre los dos grupos que formó el K-Means. El cluster 0 aglutinó mayoritariamente transacciones con valores positivos y una dispersión más amplia, mientras que el cluster 1 tuvo mayor proporción de valores negativos. Estas diferencias permitieron relacionar los grupos con diferentes tipos de movimientos comerciales, entre ellos ventas, devoluciones, rebajas o notas de crédito.

Las columnas binarias recibieron una lectura proporcional porque su media representa la frecuencia relativa de una categoría dentro del clúster. Una media cercana a uno indicó una presencia alta de la categoría dentro del grupo evaluado. Una media cercana a cero indicó una presencia reducida dentro de las transacciones correspondientes.

Los valores concretos, la cantidad de observaciones por clúster y las diferencias comerciales pertenecen al capítulo de resultados. La metodología se limitó a explicar la forma usada para construir e interpretar los grupos. Esta separación evitó mezclar el procedimiento con los hallazgos obtenidos después de ejecutar el código.

Las diferencias observadas en Total USD entre ambos clústeres deben interpretarse considerando que esta variable formó parte de las características utilizadas durante el proceso de agrupamiento, por lo que intervino directamente en la construcción de la segmentación obtenida.

9.7. Fundamento matemático del método K-Means

9.7.1. La media como representante de los datos

Perucha Jurjo (2022) parte de la media como el representante que minimiza la dispersión cuadrática de un conjunto. Sea $(C = x_1, \dots, x_n)$ un conjunto de puntos dentro de R^d, con pesos no negativos cuya suma equivale a uno. La media ponderada del conjunto se expresa mediante la siguiente relación:

$$m = \sum_{i=1}^{n} w_i x_i$$

La propiedad principal establece que esta media minimiza la suma ponderada de las distancias cuadráticas respecto de todos los puntos. Esta propiedad justifica el uso del promedio como centroide de cada grupo formado por el algoritmo. La formulación matemática correspondiente queda expresada de la siguiente forma:

$$m = \arg\min_{a \in R^d} \sum_{i=1}^{n} w_i \, \|x_i - a\|^2$$

La igualdad muestra que ningún otro punto produce una dispersión cuadrática menor para ese conjunto particular. El centroide resume las observaciones asignadas y ocupa una posición central según el criterio de mínimos cuadrados. Esta idea sirve como punto de partida para extender un representante hacia varios representantes simultáneos.

9.7.2. Extensión hacia k representantes

K-Means amplía el concepto anterior porque busca un conjunto de k representantes dentro del espacio de características. Cada observación se asocia con el representante más cercano según la distancia euclidiana cuadrática. El problema muestral puede formularse mediante la siguiente expresión:

$$H = \arg\min_{G \in B_k} \sum_{i=1}^{n} w_i \min_{g \in G\setminus} \|x_i - g\|^2$$

En esta ecuación, B_k representa la clase de conjuntos formados por k puntos dentro de R^d. El conjunto H contiene los centroides que ofrecen la menor dispersión total frente a las observaciones disponibles. Cada centroide representa un grupo cuyo contenido presenta mayor semejanza interna respecto de las características evaluadas.

Para una partición $C = C_1, C_2, \dots, C_k$, la función objetivo también puede escribirse como suma de cuadrados dentro de los grupos. El algoritmo busca una partición y unos centroides que reduzcan dicha función hasta un mínimo local. La forma usual de la función es la siguiente:

$$J(C, \mu) = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| x_i - \mu_j \right\|^2$$

9.7.3. Aplicación del criterio de representación al estudio comercial

La extensión hacia varios representantes resulta adecuada cuando una sola media no resume correctamente toda la variedad presente en los datos. Las transacciones comerciales pueden diferir por volumen, producto, ciudad, precio, descuento y tipo de pedido. Varios centroides ofrecen una descripción más precisa que un único promedio aplicado sobre todos los registros.

Dentro de esta investigación, cada centroide representó un patrón general presente en las transacciones analizadas. Los registros próximos al mismo centroide formaron un grupo con rasgos comerciales semejantes. Esta interpretación relacionó el criterio matemático de mínimos cuadrados con la estructura observada en las ventas empresariales.

9.7.4. Formulación mediante una distribución de probabilidad

El planteamiento también puede extenderse desde una muestra finita hacia una variable aleatoria multidimensional. Sea X un vector aleatorio definido sobre un espacio probabilístico y sea P la distribución inducida dentro de R^d. La esperanza matemática representa el valor medio de cada componente del vector.

La esperanza actúa como el representante que minimiza la distancia cuadrática esperada frente a los posibles valores de X. Esta propiedad generaliza la media muestral hacia una distribución de probabilidad completa. La relación matemática se expresa mediante la siguiente igualdad:

$$E(X) = \arg\min_{a \in R^d} \int \|x_i - a\|^2 dP(x)$$

La búsqueda de k representantes dentro de una distribución sigue el mismo criterio general. El problema consiste en encontrar un conjunto H que reduzca la distancia esperada hacia el representante más cercano. La formulación probabilística correspondiente queda expresada de la siguiente manera:

$$H = \arg\min_{G \in B_k} \int \min_{g \in G \setminus} \|x_i - g\|^2 dP(x)$$

9.7.5. Media phi y k-medias phi

Perucha Jurjo (2022) amplía el criterio de distancia mediante una función ϕ continua, creciente y con $\phi(0) = 0$. Esta función permite medir la diferencia entre una observación y un representante mediante una forma distinta de la distancia cuadrática. La media ϕ de una variable aleatoria se define mediante la siguiente expresión:

$$m = \arg\min_{a \in R^d} \int \phi(\|x - a\|)\ dP(x)$$

La extensión hacia k representantes recibe el nombre de k-medias ϕ. El conjunto buscado minimiza la penalización esperada entre cada observación y el representante más próximo. Su función general se expresa mediante la relación siguiente:

$$W_\phi^k(H, P) = \int \min_{h \in H} \phi(\|x - h\|)\ dP(x)$$

K-Means corresponde al caso donde $\phi(r) = r^2$, porque la penalización usa la distancia euclidiana elevada al cuadrado. Esta elección conduce al criterio de suma de cuadrados empleado por el algoritmo dentro del script. La formulación general muestra que K-Means pertenece a una familia más amplia de problemas de representación.

9.7.6. Partición del espacio mediante centroides

Sea $H = h_1, \ldots, h_k$ un conjunto de centroides obtenido mediante el criterio anterior. Cada centroide define una región formada por los puntos cuya distancia hacia él resulta menor o igual que hacia los demás representantes. La región correspondiente al centroide h_i puede escribirse de la forma siguiente:

$$C_i = x \in R^d : \|x - h_i\| \leq \|x - h_j\|,\ j \neq i$$

Estas regiones forman una partición del espacio de características y asignan cada observación a un solo grupo. Cuando un punto presenta la misma distancia hacia dos centroides, el procedimiento necesita una regla fija para resolver la asignación. El algoritmo conserva una etiqueta única para cada registro durante toda la evaluación posterior.

9.7.7. Existencia de una solución

La existencia de una solución requiere condiciones que impidan valores infinitos dentro de la función objetivo. Perucha Jurjo (2022) establece continuidad, crecimiento de (ϕ), valor nulo en el origen y existencia de algún conjunto con potencial finito. Bajo estas condiciones, existe al menos un conjunto de k representantes que alcanza el valor mínimo.

Para el caso euclidiano cuadrático, una base finita con valores numéricos válidos satisface las condiciones necesarias para ejecutar el problema muestral. Los centroides quedan dentro del espacio generado por las observaciones asignadas a cada grupo. Esta propiedad respalda la búsqueda computacional desarrollada por el algoritmo sobre la matriz estandarizada.

9.7.8. Pasos matemáticos del algoritmo iterativo

La etapa de asignación identifica el centroide más cercano para cada observación disponible. La regla compara la distancia cuadrática frente a todos los representantes existentes durante esa iteración. La asignación del registro x_i se expresa mediante la siguiente relación:

$$c_i = \arg \min_{j \in \{1,\ldots,k\}} \left\| x_i - \mu_j \right\|^2$$

Después de formar los grupos, la etapa de actualización calcula nuevamente el centroide de cada conjunto. El nuevo centroide corresponde a la media aritmética de todas las observaciones asignadas al grupo respectivo. La actualización se representa mediante la fórmula siguiente:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

La alternancia entre asignación y actualización reduce o mantiene la función objetivo durante cada iteración. El proceso termina cuando las etiquetas dejan de cambiar o cuando el algoritmo alcanza su criterio interno de parada. La solución obtenida representa un mínimo local, razón por la cual el script comparó diez inicializaciones diferentes.

9.8. Preparación de los datos para la regresión lineal

9.8.1. Construcción de variables temporales

La preparación predictiva transformó Fecha de factura en componentes numéricos aptos para la regresión. La fecha permitió obtener año, mes, día y día semanal para cada

transacción. Después de esta transformación, la fecha original se retiró de la matriz predictora.

La columna Mes existente en la base se comparó con el mes derivado desde Fecha de factura. Cuando ambas columnas representaron la misma información, se conservó una sola versión para evitar duplicidad. Esta decisión redujo relaciones repetidas dentro de la matriz usada por la regresión.

9.8.2. Definición de la respuesta y los predictores

El Total USD fue la variable dependiente porque era el valor monetario que debía estimar el modelo. La matriz de predictores incluyó variables comerciales disponibles dentro de los registros analizados. Dentro de la regresión lineal múltiple general se utilizó la etiqueta Clúster como una variable explicativa adicional, la cual representaba al grupo previamente asignado a cada transacción mediante K-Means. Para ello, el modelo de regresión se ajustó sobre el conjunto de observaciones de ambos clústeres y no por regresiones independientes para cada grupo.

Los identificadores de cliente, pedido, factura, vendedor y ruta quedaron fuera de la regresión. Kilos se conservó como medida principal de volumen y TM se retiró por su relación exacta con esa variable. Las columnas binarias derivadas de categorías se convirtieron a valores enteros de cero y uno.

9.8.3. División de entrenamiento y prueba

El script separó la base en un conjunto de entrenamiento equivalente al ochenta por ciento y otro conjunto de prueba equivalente al veinte por ciento. train_test_split realizó la partición con random_state igual a 42 para conservar el mismo resultado en nuevas ejecuciones. Se usó el conjunto de entrenamiento para la estimación de los

coeficientes, reservando el conjunto de prueba únicamente para evaluar las predicciones obtenidas. La unidad de análisis fue la transacción; de este modo, la partición train-test permitió evaluar la capacidad del modelo sobre registros no utilizados en el ajuste. El procedimiento no debía interpretarse como un pronóstico de serie temporal para meses futuros completos. Esta precisión permitió mantener la coherencia entre la estructura transaccional de la base de datos y la evaluación calculada por el script.

9.9. Entrenamiento y evaluación de la regresión lineal múltiple

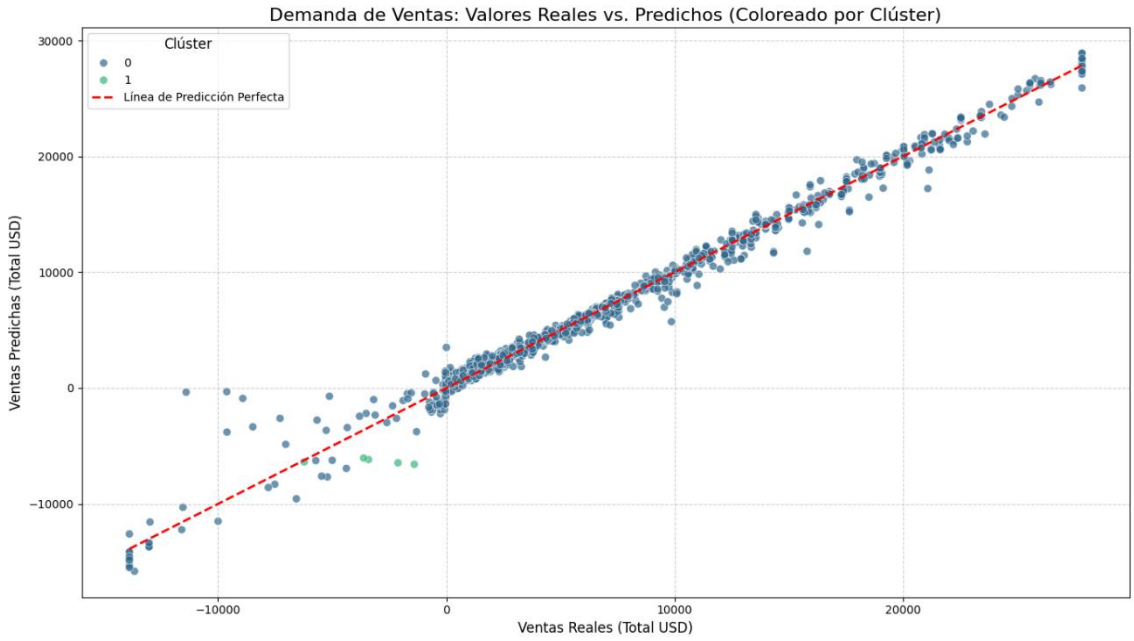
9.9.1. Ajuste y generación de predicciones

James et al. (2023) explican que la regresión lineal estima una respuesta cuantitativa a partir de uno o varios predictores. El objeto LinearRegression recibió la matriz de entrenamiento y el vector con valores observados de Total USD. El ajuste calculó un intercepto y un coeficiente para cada variable explicativa disponible.

Después del entrenamiento, el modelo generó predicciones sobre el conjunto de prueba reservado previamente. Cada valor estimado se comparó con el Total USD observado dentro de la misma transacción. Esta comparación permitió cuantificar el error del modelo sobre datos ajenos al ajuste inicial.

Figura 9

Valores reales y predichos de Total USD según el clúster



Nota. La línea diagonal representa la coincidencia perfecta entre los valores observados y las predicciones del modelo. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

La proximidad de los puntos a la diagonal permitió examinar visualmente la correspondencia entre los valores reales y las predicciones. La mayoría de los registros se encontraban cerca de la línea de referencia, aunque se observaron desviaciones en algunas transacciones con valores negativos y en algunos de los montos más altos. La diferenciación por clúster permitió identificar a qué grupo pertenecía cada una de las observaciones evaluadas.

9.9.2. Revisión de los coeficientes

Los coeficientes permitieron estudiar la dirección y la magnitud de la relación lineal existente con cada predictor. En particular, un coeficiente positivo indicaba un aumento esperado de Total USD al incrementarse la variable, manteniéndose constantes las demás; un coeficiente negativo indicaba, bajo la misma condición estadística, una reducción esperada.

La magnitud absoluta sirvió para ordenar las variables dentro del modelo, mientras que la lectura que se realizaba de la misma dependía de la escala y la codificación aplicadas. Las variables binarias explicaban el cambio respecto de una categoría que se había tomado como referencia. Esta revisión apoyó la interpretación de los resultados con fines comerciales sin atribuir relaciones causales directas a las asociaciones estimadas.

9.9.3. Métricas de evaluación

Chicco et al. (2021) también sugieren interpretar diversas métricas porque cada medida proporciona una perspectiva distinta del error predictivo. Para el estudio se calcularon el coeficiente de determinación, el error cuadrático medio y el error absoluto medio sobre el conjunto de prueba. La lectura conjunta permitió evaluar el ajuste, la magnitud promedio del error, así como la presencia de desviaciones amplias.

El error cuadrático medio elevó cada diferencia al cuadrado antes de calcular el promedio. Este cálculo atribuyó un peso mayor a las predicciones que presentaban desviaciones amplias respecto del valor observado. La ecuación aplicada fue la siguiente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

El error absoluto medio calculó el promedio de las diferencias absolutas entre valores observados y estimados. Su resultado mantuvo las unidades monetarias de Total USD y posibilitó su interpretación comercial directa. La expresión aplicada fue la siguiente:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

El coeficiente de determinación midió la proporción de variación observada que fue explicada por la regresión. Un valor mayor representó una cercanía más alta entre las predicciones y la estructura de los datos evaluados. La relación matemática correspondiente fue la siguiente:

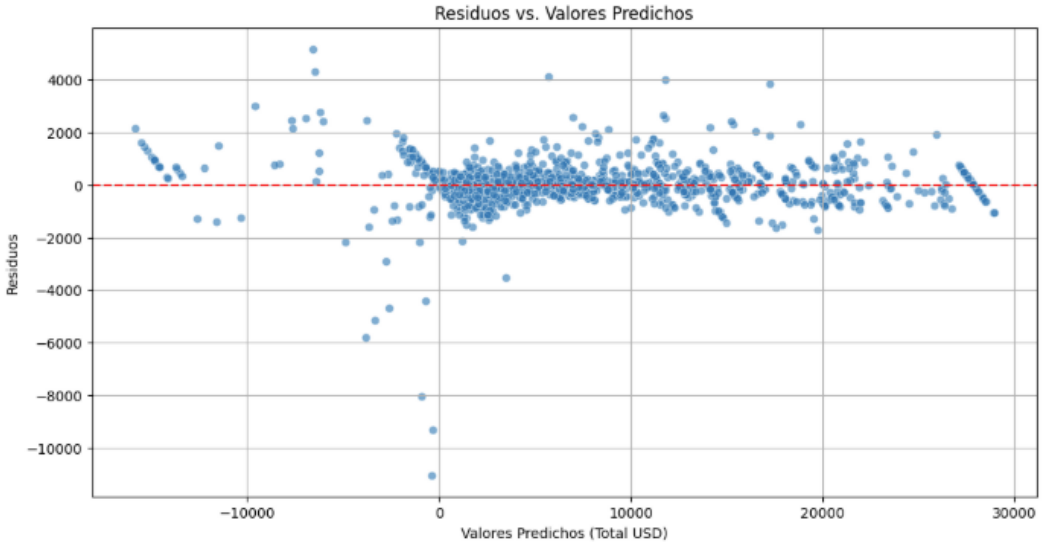
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

9.9.4. Análisis de residuos

Los residuos se calcularon como la diferencia entre Total USD observado y su valor estimado por la regresión. El gráfico de residuos frente a predicciones permitió revisar patrones, cambios de dispersión y posibles relaciones no explicadas. El histograma permitió examinar la forma general de la distribución de los errores.

Figura 10

Residuos frente a valores predichos del modelo de regresión lineal

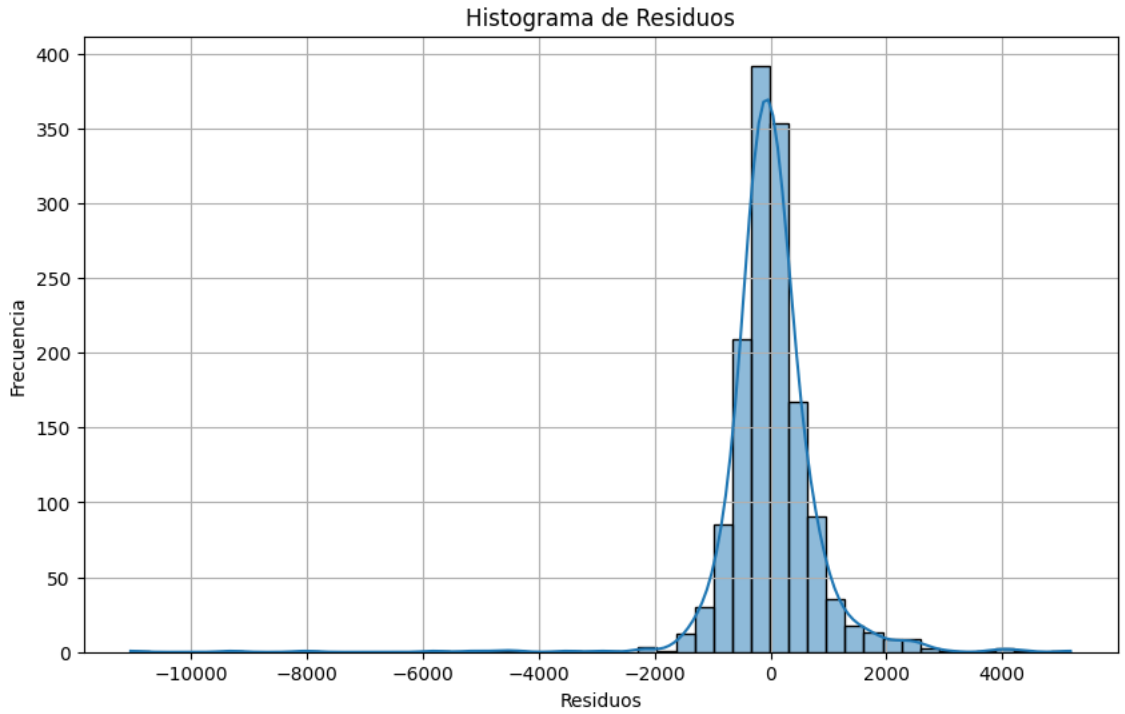


Nota. La línea horizontal ubicada en cero representa la ausencia de diferencia entre el valor observado y el valor estimado. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

En los residuos, se observó una dispersión considerable alrededor de la línea horizontal correspondiente a cero, aunque la distribución no fue completamente uniforme en todo el rango de predicciones. También se identificaron residuos alejados del grupo principal, especialmente en aquellas transacciones donde las predicciones fueron negativas o se encontraron próximas a cero. Al interpretar la estabilidad del modelo y las posibles desviaciones en operaciones poco frecuentes, estos comportamientos deben considerarse dentro del análisis.

Figura 11

Distribución de los residuos del modelo de regresión lineal



Nota. El histograma presenta la frecuencia de los errores obtenidos mediante la diferencia entre los valores observados y los valores predichos. Elaboración propia con datos de la Empresa ABC procesados en Python mediante Google Colab.

El histograma mostró una concentración importante de residuos alrededor de cero, lo que significa que una gran proporción de las predicciones presentó errores de baja

magnitud. Sin embargo, la distribución también mostró colas largas y algunas observaciones alejadas del centro, generalmente hacia los valores negativos. Por lo tanto, la evaluación del modelo se complementó con el error cuadrático medio, el error absoluto medio y el coeficiente de determinación.

Una nube de residuos sin una estructura definida alrededor de cero respaldó la presencia de una relación lineal razonable dentro del rango analizado. Un incremento progresivo de la dispersión podría indicar una posible falta de constancia en la varianza de los errores. Estas revisiones complementaron las métricas y ofrecieron una lectura gráfica del comportamiento del modelo.

9.10. Fundamento matemático de la regresión lineal múltiple

9.10.1. Modelo general

Montgomery et al. (2021) presentan la regresión lineal múltiple como una relación entre una respuesta, varios predictores y un término aleatorio. Para una observación (t), la forma general del modelo puede expresarse mediante la siguiente ecuación:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_3 X_{3t} + \cdots + \beta_k X_{kt} + u_t$$

En esta expresión, Y_t representa Total USD para la observación correspondiente. Los términos $X_{2t}, \ldots, X_{kt}$ representan las variables explicativas y u_t representa el error no observado. Los parámetros $\beta_1, \ldots, \beta_k$ cuantifican la relación lineal entre cada predictor y la respuesta.

9.10.2. Representación matricial

La notación matricial reúne todas las observaciones dentro de una sola relación algebraica. El vector y contiene los valores observados de Total USD y la matriz X

contiene los predictores seleccionados. El vector β reúne los parámetros desconocidos y el vector u representa las perturbaciones aleatorias:

$$y = X\beta + u$$

El modelo ajustado reemplaza el vector desconocido por los coeficientes calculados desde los datos de entrenamiento. Los valores estimados se obtienen mediante el producto entre la matriz predictora y el vector estimado. La expresión correspondiente queda definida de la siguiente forma:

$$\hat{y} = X\hat{\beta}$$

9.10.3. Residuos y criterio de mínimos cuadrados

El residuo representa la diferencia entre el valor observado y la predicción correspondiente. El vector residual permite medir el error cometido por el modelo sobre todas las observaciones evaluadas. Su representación matricial se expresa mediante la siguiente relación:

$$\hat{u} = y - \hat{y} = y - X\hat{\beta}$$

El método de mínimos cuadrados selecciona los coeficientes que reducen la suma de los residuos elevados al cuadrado. La función objetivo puede escribirse mediante productos matriciales para facilitar su derivación. La expresión general queda representada de la siguiente forma:

$$S(\beta) = (y - X\beta)'(y - X\beta)$$

El desarrollo algebraico de esta función produce una forma compuesta por productos entre la respuesta, los predictores y los coeficientes. El punto mínimo se obtiene

después de derivar la función respecto del vector β. La condición matemática de primer orden conduce directamente al sistema conocido como ecuaciones normales.

9.10.4. Ecuaciones normales y estimador

Las ecuaciones normales expresan la condición que deben cumplir los coeficientes de mínimos cuadrados. El sistema relaciona la matriz de predictores con el vector de respuestas observadas. Su forma matricial se presenta mediante la siguiente ecuación:

$$X'X\hat{\beta} = X'y$$

Cuando X'X posee inversa, el vector de coeficientes puede obtenerse mediante una solución cerrada. La existencia de la inversa requiere ausencia de relaciones lineales exactas entre las columnas predictoras. El estimador ordinario de mínimos cuadrados queda expresado de la siguiente manera:

$$\hat{\beta} = (X'X)^{-1}X'y$$

La eliminación de TM frente a Kilos y la conservación de una sola variable mensual redujeron relaciones lineales exactas dentro de la matriz. drop_first también retiró una categoría de referencia en cada grupo de variables binarias. Estas decisiones apoyaron la estimación de coeficientes con una matriz mejor condicionada.

9.10.5. Predicción de nuevas observaciones

La predicción de una nueva observación utiliza sus características comerciales y los coeficientes obtenidos durante el ajuste. Sea x_0 el vector que representa el nuevo registro después de aplicar las mismas transformaciones utilizadas sobre el conjunto de entrenamiento. Así, el valor predicho podrá ser obtenido mediante la siguiente relación:

$$\widehat{y_0} = x_0^{'}\hat{\beta}$$

La nueva observación debe presentar las mismas columnas, escalas y categorías empleadas durante el ajuste del modelo. Por tanto, no puede aparecer una categoría que no existiera durante el entrenamiento sin aplicar previamente una transformación compatible con la matriz utilizada por el modelo. Esta condición exige mantener el mismo proceso de preprocesamiento cuando el modelo reciba registros posteriores.

9.10.6. Supuestos del modelo

La regresión lineal múltiple parte de una relación lineal entre la respuesta y los predictores seleccionados. También supone errores con media cero, varianza constante y ausencia de correlación entre observaciones diferentes. La revisión de residuos aportó evidencia gráfica sobre varios de estos supuestos dentro del conjunto de prueba.

La matriz de predictores tiene que tener rango completo para evitar la multicolinealidad perfecta. Las variables deben contar con mediciones adecuadas y los parámetros tienen que estimarse de forma estable dentro del periodo analizado. Estas condiciones orientaron tanto la selección de columnas como la interpretación responsable de los resultados.

9.11. Secuencia general del procedimiento

La primera fase comprendió la carga del archivo, la inspección estructural y la descripción inicial de sus variables. La segunda fase trató ausencias, formatos, observaciones atípicas y columnas sin aporte para los modelos. La tercera fase codificó categorías, retiró relaciones repetidas y estandarizó las características destinadas a K-Means.

La cuarta fase evaluó diferentes valores de k mediante el codo y el coeficiente de silueta. La quinta fase ajustó K-Means, asignó clústeres y caracterizó las diferencias existentes entre los grupos. La sexta fase incorporó la etiqueta Clúster a la matriz predictora, definió Total USD como variable de respuesta y realizó la partición de entrenamiento y prueba.

La séptima fase ajustó la regresión lineal múltiple y produjo predicciones sobre el conjunto reservado. La octava fase calculó R^2, MSE, MAE, coeficientes y residuos para evaluar el modelo. Esta secuencia mantuvo una correspondencia directa con el orden lógico definido dentro del cuaderno de Google Colab.

9.12. Consideraciones éticas y control de reproducibilidad

La investigación utilizó registros comerciales internos, sin realizar experimentos con personas. Los datos recibieron un tratamiento confidencial y los resultados se mostraron en forma de valores consolidados. Los nombres, los códigos sensibles y las referencias comerciales identificables se excluyeron de las matrices de salida cuando existía riesgo de identificación de personas u operaciones concretas.

Los archivos tuvieron acceso restringido durante la ejecución del estudio académico. Las tablas, figuras y apéndices se diseñaron para no proporcionar información que pudiera identificar a clientes, vendedores u operaciones concretas. Este tratamiento protegió la confidencialidad acordada previamente con los responsables de la Empresa ABC.

La reproducibilidad se apoyó en un cuaderno secuencial, bibliotecas identificadas y parámetros aleatorios fijos. Cada etapa conservó el código necesario para repetir la

preparación, el agrupamiento, la regresión y la evaluación. Esta organización permitió revisar cambios posteriores sin perder la ruta aplicada sobre la base original.

10. DISCUSIÓN

Los resultados del trabajo realizado permitieron ver que la combinación de K-Means y regresión lineal múltiple permitió diferenciar patrones en las transacciones y obtener un alto nivel de ajuste predictivo. El modelo arrojó un R2 de 0,9883, un MSE de 710.081,00 y un MAE de 485,65, aunque se encontraron mayores desviaciones en algunas operaciones negativas o cercanas a cero.

En el caso del agrupamiento, el mejor resultado de K-Means fue con dos clústeres, mientras que el resto de configuraciones tuvieron valores menores del coeficiente de silueta. Las agrupaciones distinguieron principalmente operaciones con importes positivos de aquellas donde había más devoluciones, descuentos y notas de crédito. El resultado obtenido confirma que los registros mercantiles muestran conductas que no son susceptibles de una interpretación uniforme.

Tres segmentos de clientes fueron obtenidos por Aulia et al. (2026) con un coeficiente de silueta de 0,7913 lo que se considera una separación favorable de los grupos. Los autores identificaron clientes de alto, mediano y bajo nivel de compra, mientras que en la presente investigación se identificaron dos patrones, principalmente asociados al comportamiento de las transacciones. Eso indica que la estructura de los grupos depende de las propias características de los datos sobre los que se está realizando el análisis.

Respecto de la capacidad predictiva, Aulia et al. (2026) informaron un R2 de 0,6910, frente al 0,9883 encontrado en el presente estudio. Su modelo explicó alrededor

del 69,10 % de la variación del valor de compra, con un MAE de 213.183.958 rupias y un MSE alto, debido a valores extremos. En los dos estudios los valores atípicos influyeron sobre errores de predicción concretos.

También se observan resultados consistentes con Chen y Lu (2021) en la utilidad de agrupar los problemas comerciales. Los modelos propuestos KM-ELM y KM-SVR, obtuvieron menores errores que los modelos sin clustering, el modelo KM-ELM tuvo un MAPE de 0,13 % y un RMSE de 0,14, mientras que los errores por ELM fueron de 0,44 % y 0,62 respectivamente. Estos resultados coinciden con el desempeño favorable observado en el presente estudio.

Sin embargo, los resultados numéricos de los tres estudios no pueden compararse directamente, ya que difieren en las variables, las monedas, los sectores y las escalas analizados. El principal resultado coincide de forma general en que los modelos pueden identificar patrones comerciales y generar estimaciones con utilidad predictiva. En la presente investigación se encontró que el alto valor de R2 debe interpretarse en conjunto con los residuos observados y con las características propias de los registros utilizados.

Los resultados también guardan relación con los estudios comparados, en términos empresariales. Aulia y otros (2026) relacionaron la segmentación y predicción con la planificación comercial, y Chen y Lu (2021) afirmaron que una mejor estimación de la demanda puede ayudar al control de inventarios y disminuir los riesgos de exceso o escasez de inventarios. En el presente estudio las estimaciones pueden servir de apoyo para tomar decisiones sobre inventarios, compras, despachos, metas de venta y recursos financieros.

Los resultados muestran una correspondencia favorable con los trabajos de Aulia et al. (2026) y Chen y Lu (2021). Si bien difieren en la cantidad de grupos y en los índices

alcanzados, los tres trabajos demuestran que el análisis de patrones por medio de K-Means puede aportar valor a los procesos de predicción comercial. En este estudio identificamos dos grupos y el R2 de 0,9883 apoyan la utilidad del modelo dentro de las condiciones particulares de la empresa analizada.

Tabla 3

Diccionario de las variables contenidas en la base original

Resultado comparado	**Presente investigación**	**Aulia et al. (2026)**	**Chen y Lu (2021)**
Resultado de K-Means	2 clústeres	3 clústeres	Entre 3 y 5 según el modelo
Interpretación de grupos	Operaciones positivas frente a mayor presencia de movimientos negativos	Clientes de nivel alto, medio y bajo	Patrones diferenciados de demanda
Silueta	Mejor resultado con (k=2)	0,7913	No constituye la métrica principal reportada
(R^2)	0,9883	0,6910	No reportado como principal
MAE	485,65	Rp 213.183.958	No reportado

Resultado comparado	Presente investigación	Aulia et al. (2026)	Chen y Lu (2021)
MSE	710.081,00	Rp 206.204.672.226.881.088	No reportado
MAPE	No reportado	No reportado	Hasta 0,13 % en KM-ELM
RMSE	No reportado	No reportado	Hasta 0,14 en KM-ELM
Comportamiento de errores	Desviaciones en operaciones negativas o cercanas a cero	Valores extremos afectan las predicciones	Los modelos con clustering redujeron los errores
Resultado general	Alto ajuste dentro de las transacciones estudiadas	Capacidad predictiva moderada y segmentación definida	Mayor precisión de los modelos con clustering

Nota: Elaboración propia a partir de (Aulia, Priatna, & Yasir, 2024; Chen & Lu, 2021)

11. CONCLUSIONES

La investigación permitió cumplir el objetivo general al analizar la estimación de ventas mediante una combinación de K-Means y regresión lineal múltiple. El estudio se realizó con 7.216 registros y 38 variables correspondientes a transacciones comerciales de alimento balanceado para camarón. La revisión de la base permitió reconocer que el comportamiento de las ventas depende de cantidades, precios, descuentos, productos,

fechas, ciudades y tipos de operación. La aplicación de técnicas de machine learning proveyó una forma ordenada de procesar esta diversidad de información. Así, el histórico de ventas pudo convertirse en una fuente útil para describir patrones y construir estimaciones sobre la variable Total USD.

La revisión de la literatura permitió establecer que el aprendizaje automático tiene aplicaciones relevantes dentro del sector acuícola y en la gestión comercial. Los estudios consultados mostraron que las técnicas supervisadas permiten estimar variables numéricas, mientras que los métodos no supervisados permiten reconocer grupos sin categorías definidas previamente. Esta base teórica dio soporte al uso conjunto de K-Means y regresión lineal múltiple dentro de la investigación. También permitió relacionar el análisis de ventas con necesidades empresariales como la planificación de inventarios, el abastecimiento, la distribución y el control financiero. Así, el trabajo mantuvo coherencia entre el problema planteado, los objetivos y el procedimiento seguido.

Se realizó un pretratamiento de los datos para obtener una matriz adecuada antes de empezar a aplicar los modelos. Este proceso de limpieza permitió revisar valores faltantes, datos atípicos, formatos incorrectos, variables categóricas, identificadores y columnas con información repetida. La codificación One-Hot transformó las categorías en una representación numérica, mientras que la estandarización colocó las características empleadas por K-Means dentro de una escala comparable. Por último, se eliminaron variables que no representaban una medición comercial directa. Total USD se conservó como una característica utilizada durante el agrupamiento, de acuerdo con la configuración metodológica aplicada, y la etiqueta Clúster que se obtuvo más tarde se introdujo como variable explicativa dentro de la regresión lineal múltiple general.

Se evaluó el número de grupos, y el método del codo no presentó un punto de inflexión totalmente definido, por lo cual también se apoyó la decisión en el coeficiente

de silueta. Este indicador tuvo mejor resultado con dos clusters, configuración que se utilizó para ejecutar K-Means. La caracterización mostró que el clúster 0 agrupó principalmente transacciones con valores positivos y una mayor dispersión de Total USD. Los valores negativos del cluster 1 se corresponden con las devoluciones, los descuentos o las notas de crédito, y son mayores en proporción. Por lo tanto, el agrupamiento permitió diferenciar tipos de movimientos comerciales y no debe interpretarse como una segmentación directa de personas o clientes.

La regresión lineal múltiple obtuvo un coeficiente de determinación de 0,9883, un error cuadrático medio de 710.081,00 y un error absoluto medio de 485,65. Estos resultados muestran que, dentro del conjunto de prueba utilizado, las ventas observadas se aproximan de manera muy alta a las predicciones. El gráfico comparativo confirmó que la mayoría de los puntos se situó cerca de la línea de coincidencia perfecta. Sin embargo, el análisis de residuos mostró una dispersión no uniforme y observaciones alejadas del centro, especialmente en algunas transacciones negativas o cercanas a cero. Por eso, el modelo da un alto ajuste dentro de la base analizada. Este nivel de ajuste debe interpretarse considerando que la variable Clúster utilizada como predictora fue obtenida mediante un agrupamiento en el cual también participó Total USD, condición metodológica que pudo contribuir al desempeño observado.

El procedimiento desarrollado puede servir de apoyo a la toma de decisiones comerciales con la información obtenida del histórico transaccional de la empresa. Las estimaciones sirven para orientar la revisión de inventarios, compras, despachos, metas de venta y recursos financieros, pero no sustituyen el juicio de los responsables de cada área. La ejecución en Phyton mediante Google Colab facilita la repetición del análisis cuando se agreguen nuevos registros y permite mantener un trazado verificable desde la limpieza a la evaluación. En líneas generales, la combinación de K-Means y regresión

lineal múltiple fue útil para agrupar transacciones y estimar Total USD. Su uso futuro dependerá de la actualización periódica de la base y de la revisión constante del desempeño del modelo.

12. BIBLIOGRAFÍA

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 40–79. doi:https://doi.org/10.1214/09-SS054

Asamblea Nacional del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales.* Registro Oficial del Ecuador, Quito. Obtenido de https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/06/Ley-Organica-de-Datos-Personales.pdf

Aulia, T., Priatna, W., & Yasir, M. (2024). Clustering and sales prediction using K-Means and simple linear regression. *International Journal of Information Technology and Computer Science Applications*, 1–12. doi:https://doi.org/10.33395/jitcsa.v2i1.209

Aulia, T., Priatna, W., & Yasir, M. (2024). Clustering and sales prediction using K-Means and simple linear regression. *International Journal of Information Technology and Computer Science Applications*, 1–12. doi:https://doi.org/10.33395/jitcsa.v2i1.209

Aung, T., Abdul Razak, R., & Rahiman Bin Md Nor, A. (2025). Artificial intelligence methods used in various aquaculture applications: A systematic literature review. *Journal of the World Aquaculture Society*, 13107. doi:https://doi.org/10.1111/jwas.13107

BM Editores. (2022). *Cultivo global de camarón: desafíos y oportunidades.* Obtenido de https://bmeditores.mx/entorno-pecuario/cultivo-global-de-camaron-desafios-y-oportunidades/

Chen, I., & Lu, C. (2021). Demand forecasting for multichannel fashion retailers by integrating clustering and machine learning algorithms. *Processes*. doi:10.3390/pr9091578

Chicco, D., Warrens, M. J., & Jurman, G. (5 de 7 de 2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science, 7*, e623. doi:10.7717/peerj-cs.623

Confirmado. (2026). *La intensificación del cultivo del camarón en Ecuador impulsa nuevas exigencias nutricionales*. Obtenido de https://confirmado.net/la-intensificacion-del-cultivo-del-camaron-en-ecuador-impulsa-nuevas-exigencias-nutricionales/

Creswell, J. W., & Creswell, J. D. (2023). *Research design: Qualitative, quantitative, and mixed methods approaches* (6 ed.). SAGE Publications, Inc. Obtenido de https://uk.sagepub.com/en-gb/eur/research-design-international-student-edition/book280802

El Productor. (2017). *La nutrición y alimentación en la acuacultura de América Latina*. Obtenido de El Productor: https://elproductor.com/2017/10/la-nutricion-y-alimentacion-en-la-acuacultura-de-america-latina/

El Universo. (2023). *Alimentación y nutrición son los aliados en la salud del camarón ecuatoriano*. Obtenido de https://www.eluniverso.com/noticias/economia/alimentacion-y-nutricion-son-los-aliados-en-la-salud-del-camaron-ecuatoriano-nota/

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 861–874.

Gladju, J., Kamalam, B. S., & Kanagaraj, A. (2022). Applications of data mining and machine learning framework in aquaculture and fisheries: A review. *Smart Agricultural Technology, 2*, 100061. doi:10.1016/j.atech.2022.100061

Global Seafood Advocate. (2018). *La industria de cultivo de camarón en Ecuador, parte 1*. Obtenido de https://www.globalseafood.org/advocate/la-industria-de-cultivo-de-camaron-en-ecuador-parte-1/

Hartomo, K. D., & Nataliani, Y. (2 de 6 de 2021). A new model for learning-based forecasting procedure by combining k-means clustering and time series forecasting algorithms. *PeerJ Computer Science, 7*, e534. doi:10.7717/peerj-cs.534

Hartomo, K. D., & Nataliani, Y. (2021). A new model for learning-based forecasting procedure by combining K-Means clustering and time series forecasting algorithms. *PeerJ Computer Science*, e534. doi:https://doi.org/10.7717/peerj-cs.534

Hartomo, K., & Nataliani, Y. (2021). A new model for learning-based forecasting procedure by combining K-Means clustering and time series forecasting algorithms. *PeerJ Computer Science*, e534. doi:https://doi.org/10.7717/peerj-cs.534

Intriago Sánchez, N. A., & Landeta Púa, D. C. (2024). *Modelo para pronosticar la demanda de alimentos balanceados en el sector acuícola.* Escuela Superior Politécnica del Litoral (ESPOL), Facultad de Ingeniería en Electricidad y Computación (FIEC). Guayaquil: ESPOL. FIEC. Obtenido de Repositorio Institucional de la ESPOL: http://www.dspace.espol.edu.ec/handle/123456789/60502

Islam, S. M., Hossain, M. S., & Al-Kadri, M. (2024). Smart aquaculture analytics: Enhancing shrimp farming in Bangladesh through real-time IoT monitoring and predictive machine learning analysis. *Heliyon*, e37330. doi:https://doi.org/10.1016/j.heliyon.2024.e37330

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python* (1 ed.). Springer. doi:10.1007/978-3-031-38747-0

Lorente-Leyva, L. L., Herrera-Granda, I. D., Martínez-Granda, M., Pérez-Salinas, C., Herrera-Granda, E. P., Cuarán-Cuarán, J., & Acosta-Vargas, P. (2020). Demand forecasting of fashion textile products: A comparison of neural networks and classical forecasting methods in the Ecuadorian textile industry. *Mathematics*, 1612. doi:https://doi.org/10.3390/math8091612

Lorente-Leyva, L. L., Herrera-Granda, I. D., Martínez-Granda, M., Pérez-Salinas, C., Herrera-Granda, E. P., Cuarán-Cuarán, J., & Acosta-Vargas, P. (2020). Demand forecasting of fashion textile products: A comparison of neural networks and classical forecasting methods in the Ecuadorian textile industry. *Mathematics*, 1612. doi:https://doi.org/10.3390/math8091612

Majeed, A., Tahir, M. N., Khan, A. R., Zubair, M., & Hassan, S. (2025). Hybrid unsupervised-supervised learning framework for rainfall prediction using satellite signal strength attenuation. *PLoS ONE*. Obtenido de https://pmc.ncbi.nlm.nih.gov/articles/PMC12846065/

Malik, S., Khan, M., Abid, M. K., & Aslam, N. (2024). Sales forecasting using machine learning algorithm in the retail sector. *ournal of Computing & Biomedical Informatics*, 282–294. doi:https://doi.org/10.56979/401/2024

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis* (6 ed.). Wiley. Obtenido de https://www.wiley.com/en-us/Introduction+to+Linear+Regression+Analysis%2C+6th+Edition-p-9781119578758

Mujahid, M. A., Azizah, F. F., Indrayana, G. G., Rachminiwati, N., Sakai, Y., & Yagi, N. (2025). Prediction of shrimp growth by machine learning: The use of actual data of industrial-scale outdoor white shrimp (Litopenaeus vannamei) aquaculture in Indonesia. *Aquaculture Journal*, 27. doi:https://doi.org/10.3390/aquacj5040027

Olaitan, F. A., Adeniyi, A. E., & Onifade, F. S. (2024). Predicting future sales: A machine learning algorithm showdown. En M. S. Obaidat & P. Nicopolitidis (Eds.),. *Advances in Intelligent Systems and Computing*, 38–52. doi: https://doi.org/10.1007/978-3-031-48465-0_4

Peng, J., Hahn, J., & Huang, K.-W. (31 de 3 de 2022). Handling missing values in information systems research: A review of methods and assumptions. *Information Systems Research, 34*(1), 5-26. doi:10.1287/isre.2022.1104

Perucha Jurjo, C. (2022). *El método de k-medias.* Universidad de Valladolid, Facultad de Ciencias, Valladolid. Obtenido de https://uvadoc.uva.es/handle/10324/58229

Presidencia de la República del Ecuador. (2023). *Decreto Ejecutivo No. 904: Reglamento General de la Ley Orgánica de Protección de Datos Personales.* Registro Oficial del Ecuador, Quito. Obtenido de https://www.registroficial.gob.ec/tercer-suplemento-al-registro-oficial-no-435/

Primicias. (2025). *El camarón fue el primer producto de exportación de Ecuador, por encima del petróleo, en el primer semestre*. Obtenido de https://www.primicias.ec/economia/camaron-petroleo-exportaciones-semestre-ecuador-102852/

Rabia, M., Aslam, N., M., Rashid, S., & Shafique, M. A. (2024). AI-driven aquafarming: Recent innovations. Poultry. *Aquaculture & Fisheries Management for Biomedical Research*, 1–15. Obtenido de https://academicstrive.com/PAFMB/PAFMB180046.pdf

Redalyc. (2022). *Algoritmos de aprendizaje supervisado para proyección de ventas de camarón ecuatoriano con lenguaje de programación Python.* Obtenido de https://redalyc.org/journal/6955/695574853003/html/

REVISTVR. (2025). Análisis de la competitividad de las exportaciones ecuatorianas en el mercado global 2021–2023. *Revista Instituto Superior Tecnológico Vicente Rocafuerte*. Obtenido de https://revista.istvr.edu.ec/

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach (4.ª ed.). Pearson.*

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 160. doi:https://doi.org/10.1007/s42979-021-00592-x

Sarker, I. H. (22 de 3 de 2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science, 2*, 160. doi:10.1007/s42979-021-00592-x

Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 34–38. doi:https://doi.org/10.14569/IJARAI.2013.020206

Sotaquira, M. (2025). *El camarón ecuatoriano retoma impulso en 2025 tras leve caída en exportaciones en 2023 y 2024*. Obtenido de Lupa Media: https://lupa.com.ec/explicativos/exportaciones-camaron-ecuador-mundo/

Stony Brook University, Florida International University, & National Taiwan Ocean University. (2016). *El origen filogenético de los camarones que dominan la*

acuicultura mundial. Obtenido de Aquahoy: https://aquahoy.com/el-origen-filogenetico-de-los-camarones-que-dominan-la-acuicultura-mundial/

Superintendencia de Protección de Datos Personales. (2025). *Reglamento para la seudonimización, anonimización, bloqueo y eliminación de datos personales.* Superintendencia de Protección de Datos Personales, Quito. Obtenido de https://spdp.gob.ec/wp-content/uploads/2025/08/0030-R.pdf

Superintendencia de Protección de Datos Personales. (2026). *Norma General para la Garantía del Derecho de Protección de Datos Personales en el Uso de Sistemas de Inteligencia Artificial.* Superintendencia de Protección de Datos Personales, Quito. Obtenido de https://spdp.gob.ec/wp-content/uploads/2026/02/ResolIA.pdf

Sustainable Shrimp Partnership [SSP]. (2023). *El nacimiento casual de la acuicultura de camarón ecuatoriana*. Obtenido de https://sustainableshrimppartnership.org/es/el-nacimiento-casual-de-la-acuicultura-de-camaron-ecuatoriana/

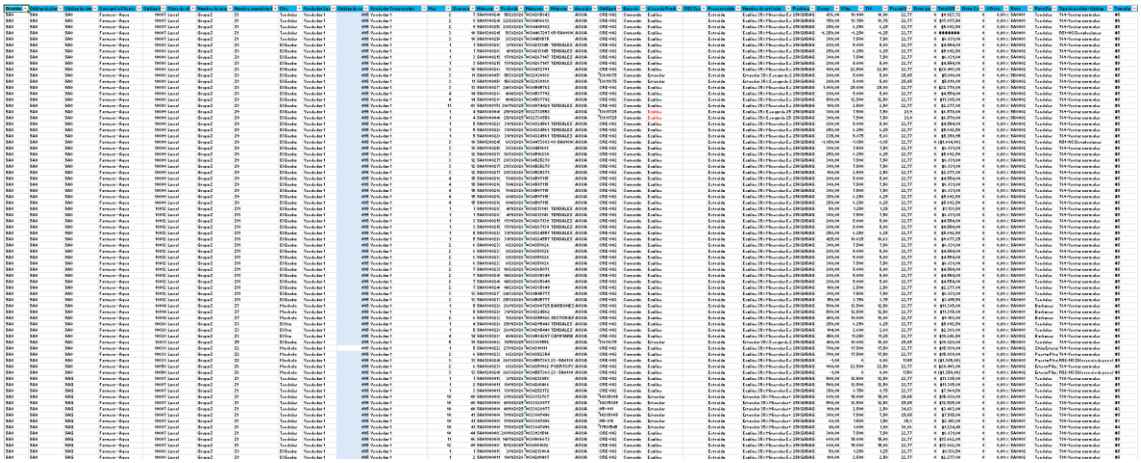
Suyal, M., & Sharma, S. (2024). A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning. *Journal of Artificial Intelligence and Systems*, 85–95. doi:https://doi.org/10.33969/AIS.2024060106

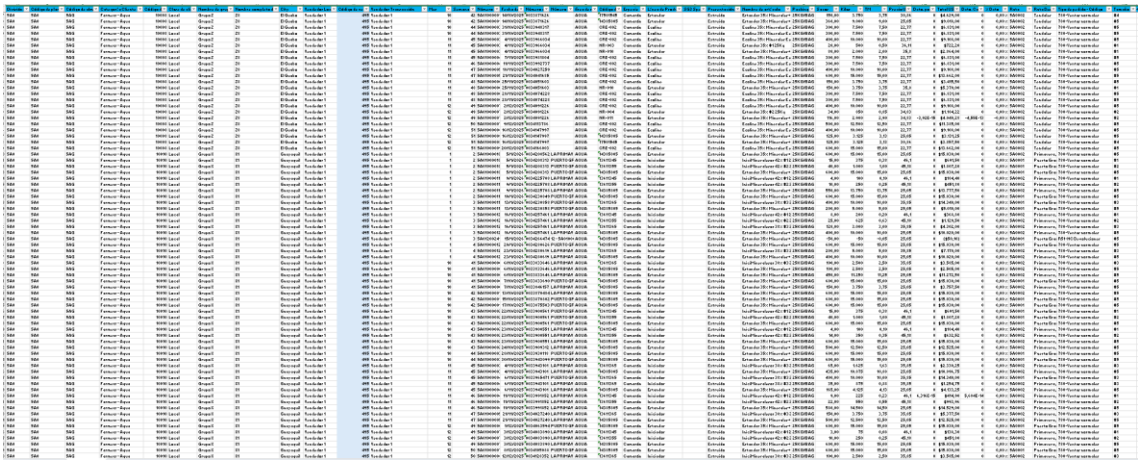
Suyal, M., & Sharma, S. (2024). A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning. *Journal of Artificial Intelligence and Systems*, 85–95. doi:https://doi.org/10.33969/AIS.2024060106

Tuyen, T. T., Al-Ansari, N., N., D. D., L. H., Phan, T. N., Prakash, I., . . . Pham, B. T. (2024). Prediction of white spot disease susceptibility in shrimps using decision trees based machine learning models. *Applied Water Science*, 2. doi:https://doi.org/10.1007/s13201-023-02049-3

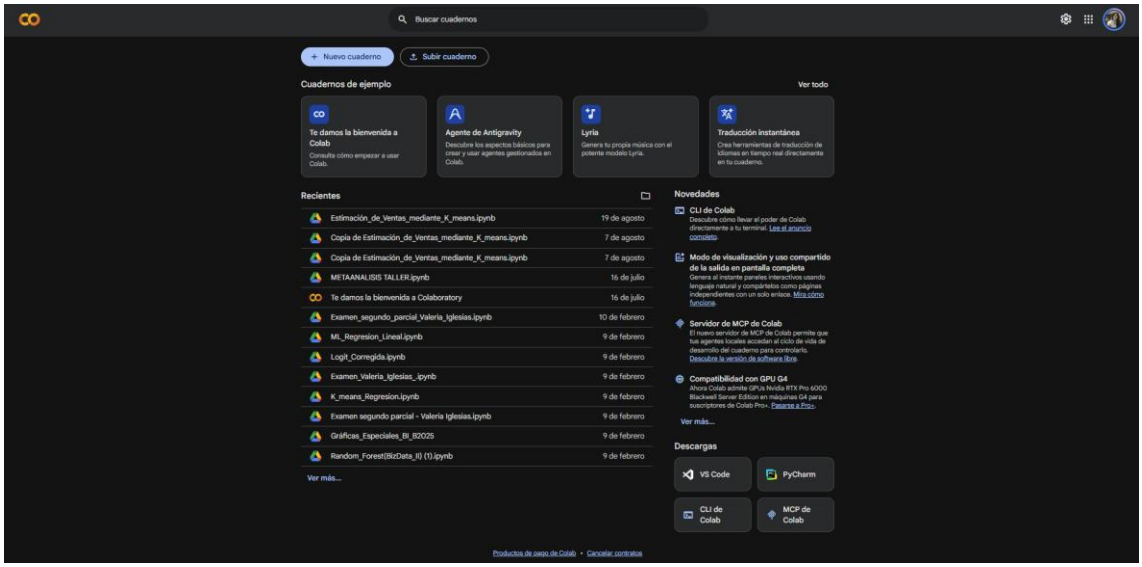
13. ANEXOS

Anexo 1 Base de datos





Anexo 2 Entorno Google Colab



Anexo 3 Script en Python

```
#Cargamos las librerias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import warnings
warnings.filterwarnings('ignore')
#Librerias para k means
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
#Librerias para la predicción
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

+ Código + Texto

```
from google.colab import files
import pandas as pd

# Upload the Excel file from your local machine
print("Please upload your 'base trabajada jose cañar e Iglesias .xlsx' file:")
uploaded = files.upload()

if uploaded:
    # Get the actual filename from the uploaded dictionary keys
    file_name = list(uploaded.keys())[0]
    df = pd.read_excel(file_name)
    print(f"'{file_name}' uploaded and loaded successfully into DataFrame 'df'.")
    display(df.head())
else:
    print("No file was uploaded.")
```

```
Please upload your 'base trabajada jose cañar e Iglesias .xlsx' file:
Elegir archivos Ningún archivo seleccionado Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving base trabajada jose cañar e Iglesias .xlsx to base trabajada jose cañar e Iglesias .xlsx
'base trabajada jose cañar e Iglesias .xlsx' uploaded and loaded successfully into DataFrame 'df'.
```

	División	Código de planta	Código de almacén	Categoría Cliente	Código de cliente	Clase de cliente	Nombre de grupo	Nombre completo de cliente	City	Vendedor Local Asignado	...	TM	Precio Unit.	Dcto. por Saco	Total USD	Dcto. Calculado	% Dcto	Ruta	Ruta Descripción	Tipo de pedido - Código	Tamaño Partícula
0	5A0	5A0	5A0	Farmers - Aqua	10007.0	Local	Grupo Z	Z1	Tendales	Vendedor 1	...	10.90	22.77	0.0	9927.72	0.0	0.0	5A0001	Tendales	769-Ventas normales	#5
1	5A0	5A0	5A0	Farmers - Aqua	10007.0	Local	Grupo Z	Z1	Tendales	Vendedor 1	...	18.75	22.77	0.0	17077.50	0.0	0.0	5A0001	Tendales	769-Ventas normales	#5
2	5A0	5A0	5A0	Farmers - Aqua	10007.0	Local	Grupo Z	Z1	Tendales	Vendedor 1	...	6.25	22.77	0.0	5692.50	0.0	0.0	5A0001	Tendales	769-Ventas normales	#5
3	5A0	5A0	5A0	Farmers - Aqua	10007.0	Local	Grupo Z	Z1	Tendales	Vendedor 1	...	-6.25	22.77	0.0	-5692.50	0.0	0.0	5A0001	Tendales	R51-NC Devoluciones	#5
4	5A0	5A0	5A0	Farmers - Aqua	10007.0	Local	Grupo Z	Z1	Tendales	Vendedor 1	...	7.50	22.77	0.0	6831.00	0.0	0.0	5A0001	Tendales	769-Ventas normales	#5

5 rows × 38 columns

1. Inspección Inicial del DataFrame

Primero, veamos la información general del DataFrame, incluyendo los tipos de datos y la cantidad de valores no nulos, para identificar columnas numéricas y detectar posibles valores faltantes.

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Data columns (total 38 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   División                        7215 non-null   object
 1   Código de planta                7213 non-null   object
 2   Código de almacén               7213 non-null   object
 3   Categoría Cliente               7213 non-null   object
 4   Código de cliente               7213 non-null   float64
 5   Clase de cliente                7213 non-null   object
 6   Nombre de grupo                 7213 non-null   object
 7   Nombre completo de cliente      7213 non-null   object
 8   City                            7213 non-null   object
 9   Vendedor Local Asignado         7213 non-null   object
 10  Código de vendedor - Transacción 7213 non-null  float64
 11  Vendedor Transacción            7213 non-null   object
 12  Mes                             7213 non-null   float64
 13  Semana                          7213 non-null   float64
 14  Número de factura               7213 non-null   object
 15  Fecha de factura                7213 non-null   datetime64[ns]
 16  Número de pedido                7213 non-null   float64
 17  Número de pedido del cliente    4632 non-null   object
 18  Área de negocio                 7213 non-null   object
 19  Código de artículo              7213 non-null   object
 20  Especie                         7213 non-null   object
 21  Línea de Producto               7213 non-null   object
 22  SG3 Species L4                  0 non-null      float64
 23  Presentación de Producto        7213 non-null   object
 24  Nombre de artículo              7213 non-null   object
 25  Packing Sales                   7213 non-null   object
 26  Sacos                           7214 non-null   float64
 27  Kilos                           7214 non-null   float64
 28  TM                              7214 non-null   float64
 29  Precio Unit.                    7213 non-null   float64
 30  Dcto. por Saco                  7209 non-null   float64
 31  Total USD                       7214 non-null   float64
 32  Dcto. Calculado                 7210 non-null   float64
 33  % Dcto                          7209 non-null   float64
 34  Ruta                            7212 non-null   object
 35  Ruta Descripción                7212 non-null   object
 36  Tipo de pedido - Código         7213 non-null   object
 37  Tamaño Partícula                7175 non-null   object
dtypes: datetime64[ns](1), float64(14), object(23)
memory usage: 2.1+ MB
None
```

```
display(df.describe())
```

	Código de cliente	Código de vendedor - Transacción	Mes	Semana	Fecha de factura	Número de pedido	SG3 Species L4	Sacos	Kilos	TN	Precio Unit.	Dcto. por Saco	Total USD	Dcto. Calculado	% Dcto
count	7213.000000	7213.0	7213.000000	7213.000000	7213	7.213000e+03	0.0	7214.000000	7.214000e+03	7214.000000	7213.000000	7.209000e+03	7.214000e+03	7210.000000	7.209000e+03
mean	10034.544156	495.0	5.730625	23.623319	2025-10-02 12:14:40.410369792	3.380879e+07	NaN	266.271694	1.397212e+04	13.972124	85.648157	1.932554e-01	1.527753e+04	137.938025	6.188068e-03
min	10001.000000	495.0	1.000000	1.000000	2025-01-02 00:00:00	3.262196e+07	NaN	-60000.000000	-6.000000e+04	-60.000000	0.230000	-1.136868e-14	-6.408000e+04	-1836.000000	-2.153772e-16
25%	10025.000000	495.0	3.000000	11.000000	2025-06-07 00:00:00	3.331674e+07	NaN	50.000000	1.250000e+03	1.250000	26.000000	0.000000e+00	1.765625e+03	0.000000	0.000000e+00
50%	10035.000000	495.0	5.000000	19.000000	2025-10-17 00:00:00	3.388106e+07	NaN	200.000000	5.000000e+03	5.000000	29.450000	0.000000e+00	5.664000e+03	0.000000	0.000000e+00
75%	10045.000000	495.0	9.000000	38.000000	2026-01-30 00:00:00	3.431502e+07	NaN	450.000000	1.125000e+04	11.250000	36.000000	0.000000e+00	1.220000e+04	0.000000	0.000000e+00
max	10071.000000	495.0	12.000000	53.000000	2026-05-20 00:00:00	3.478643e+07	NaN	960442.000000	5.039745e+07	50397.450000	15013.950000	1.058000e+01	5.510604e+07	497266.580000	2.177851e-01
std	15.607033	0.0	3.568628	15.484367	NaN	5.987568e+05	NaN	11420.284410	5.933378e+05	593.337809	661.198989	5.878275e-01	6.487689e+05	5861.179529	1.839686e-02

2. Manejo de Valores Faltantes (Missing Values)

Identificaremos las columnas con valores faltantes y decidiremos una estrategia para tratarlos. Para columnas numéricas, consideraremos imputar con la mediana o media, o eliminarlas si tienen muchos valores faltantes. Para categóricas, podemos imputar con la moda o crear una categoría 'Desconocido'.

```
missing_values = df.isnull().sum()
missing_values = missing_values[missing_values > 0].sort_values(ascending=False)
print("Valores faltantes por columna:")
print(missing_values)

# Estrategia de imputación:
# 1. 'Código de cliente' y 'TN' parecen numéricas. Imputaremos con la mediana.
# 2. Las columnas con muchos NaNs como 'División', 'Código de planta', 'Código de almacén' o 'Categoría' podrían ser categóricas.
#    Para identificadores o categorías, podríamos imputar con la moda o un valor 'Desconocido'.
#    Basado en el df.info() previo, 'Código de cliente' es float y tiene NaNs.
#    'TN' también tiene NaNs y es numérica.

# Imputar 'Código de cliente' y 'TN' con la mediana
if 'Código de cliente' in df.columns:
    df['Código de cliente'] = df['Código de cliente'].fillna(df['Código de cliente'].median())
if 'TN' in df.columns:
    df['TN'] = df['TN'].fillna(df['TN'].median())

# Para otras columnas con NaNs que pudieran ser categóricas y no aparecieron en la muestra
# pero 'df.info()' mostró, podríamos rellenarlas con la moda o 'Desconocido'.
# Ejemplo si 'Vendedor Local Asignado' tuviera NaNs:
# if 'Vendedor Local Asignado' in df.columns and df['Vendedor Local Asignado'].isnull().any():
#     df['Vendedor Local Asignado'] = df['Vendedor Local Asignado'].fillna(df['Vendedor Local Asignado'].mode()[0])
```

```
#     df['Vendedor Local Asignado'] = df['Vendedor Local Asignado'].fillna(df['Vendedor Local Asignado'].mode()[0])

print("\nValores faltantes después de la imputación:")
print(df.isnull().sum()[df.isnull().sum() > 0])
```

```
Valores faltantes por columna:
SG3 Species L4                  7216
Número de pedido del cliente    2584
Tamaño Partícula                  41
Dcto. por Saco                     7
% Dcto                             7
Dcto. Calculado                    6
Ruta Descripción                   4
Ruta                               4
Línea de Producto                  3
Clase de cliente                   3
Código de almacén                  3
Código de planta                   3
Categoría Cliente                  3
Código de cliente                  3
Semana                             3
Nombre de grupo                    3
City                               3
Nombre completo de cliente         3
Código de vendedor - Transacción   3
Vendedor Transacción               3
Mes                                3
Vendedor Local Asignado            3
Packing Sales                      3
Especie                            3
Código de artículo                 3
Área de negocio                    3
Número de pedido                   3
Fecha de factura                   3
Número de factura                  3
Presentación de Producto           3
Tipo de pedido - Código            3
Nombre de artículo                 3
Precio Unit.                       3
TN                                 2
Sacos                              2
Kilos                              2
Total USD                          2
División                           1
dtype: int64

Valores faltantes después de la imputación:
División                        1
Código de planta                3
Código de almacén               3
Categoría Cliente               3
Clase de cliente                3
Nombre de grupo                 3
Nombre completo de cliente      3
City                            3
Vendedor Local Asignado         3
Código de vendedor - Transacción 3
Vendedor Transacción            3
Mes                             3
Semana                          3
Número de factura               3
Fecha de factura                3
```

ariables | Terminal | Python 3

3. Conversión de Tipos de Datos

Convertiremos `Código de cliente` a tipo entero, ya que es un identificador.

```
# Convertir 'Código de cliente' a tipo entero
if 'Código de cliente' in df.columns:
    df['Código de cliente'] = df['Código de cliente'].astype(int)

print("Tipos de datos después de la conversión:")
print(df.info())
```

```
Tipos de datos después de la conversión:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Data columns (total 38 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   División                        7215 non-null   object
 1   Código de planta                7213 non-null   object
 2   Código de almacén               7213 non-null   object
 3   Categoría Cliente               7213 non-null   object
 4   Código de cliente               7216 non-null   int64
 5   Clase de cliente                7213 non-null   object
 6   Nombre de grupo                 7213 non-null   object
 7   Nombre completo de cliente      7213 non-null   object
 8   City                            7213 non-null   object
 9   Vendedor Local Asignado         7213 non-null   object
 10  Código de vendedor - Transacción 7213 non-null   float64
 11  Vendedor Transacción            7213 non-null   object
 12  Mes                             7213 non-null   float64
 13  Semana                          7213 non-null   float64
 14  Número de factura               7213 non-null   object
 15  Fecha de factura                7213 non-null   datetime64[ns]
 16  Número de pedido                7213 non-null   float64
 17  Número de pedido del cliente    4632 non-null   object
 18  Área de negocio                 7213 non-null   object
 19  Código de artículo              7213 non-null   object
 20  Especie                         7213 non-null   object
 21  Línea de Producto               7213 non-null   object
 22  SG3 Species L4                  0 non-null      float64
 23  Presentación de Producto        7213 non-null   object
 24  Nombre de artículo              7213 non-null   object
 25  Packing Sales                   7213 non-null   object
 26  Sacos                           7214 non-null   float64
 27  Kilos                           7214 non-null   float64
 28  TN                              7216 non-null   float64
 29  Precio Unit.                    7213 non-null   float64
 30  Dcto. por Saco                  7209 non-null   float64
 31  Total USD                       7214 non-null   float64
 32  Dcto. Calculado                 7210 non-null   float64
 33  % Dcto                          7209 non-null   float64
 34  Ruta                            7212 non-null   object
 35  Ruta Descripción                7212 non-null   object
 36  Tipo de pedido - Código         7213 non-null   object
 37  Tamaño Partícula                7175 non-null   object
dtypes: datetime64[ns](1), float64(13), int64(1), object(23)
```

4. Detección y Manejo de Outliers (Valores Atípicos) usando el Rango Intercuartílico (IQR)

Aplicaremos el método IQR a las columnas numéricas para identificar y manejar los valores atípicos. Sustituiremos los valores atípicos por los límites superior o inferior (capping) para evitar que distorsionen el análisis o los modelos.

```
numeric_cols = df.select_dtypes(include=np.number).columns.tolist()

print("Columnas numéricas para el tratamiento de outliers:", numeric_cols)

for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Identificar outliers
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    if not outliers.empty:
        print(f"\nColumna '{col}': {len(outliers)} outliers detectados.")
        # Capping: Sustituir outliers por los límites
        df[col] = np.where(df[col] < lower_bound, lower_bound, df[col])
        df[col] = np.where(df[col] > upper_bound, upper_bound, df[col])
        print(f"Outliers en '{col}' tratados mediante capping (reemplazados por límites).")
    else:
        print(f"\nColumna '{col}': No se detectaron outliers significativos.")

print("\nLimpieza de outliers completada para las columnas numéricas.")

# Mostrar el resumen estadístico después de tratar los outliers
print("\nResumen estadístico después de tratar outliers:")
display(df.describe())
```

```
Columnas numéricas para el tratamiento de outliers: ['Código de cliente', 'Código de vendedor - Transacción', 'Mes', 'Semana', 'Número de pedido', 'SG3 Species L4', 'Sacos', 'Kilos', 'TM', 'Precio Unit.', 'Dcto. por Saco', 'Total USD', 'Dcto. Calculado', '% Dcto']

Columna 'Código de cliente': No se detectaron outliers significativos.

Columna 'Código de vendedor - Transacción': No se detectaron outliers significativos.

Columna 'Mes': No se detectaron outliers significativos.

Columna 'Semana': No se detectaron outliers significativos.

Columna 'Número de pedido': No se detectaron outliers significativos.

Columna 'SG3 Species L4': No se detectaron outliers significativos.

Columna 'Sacos': 373 outliers detectados.
Outliers en 'Sacos' tratados mediante capping (reemplazados por límites).

Columna 'Kilos': 228 outliers detectados.
Outliers en 'Kilos' tratados mediante capping (reemplazados por límites).

Columna 'TM': 228 outliers detectados.
Outliers en 'TM' tratados mediante capping (reemplazados por límites).

Columna 'Precio Unit.': 321 outliers detectados.
Outliers en 'Precio Unit.' tratados mediante capping (reemplazados por límites).

Columna 'Dcto. por Saco': 1468 outliers detectados.
Outliers en 'Dcto. por Saco' tratados mediante capping (reemplazados por límites).

Columna 'Total USD': 249 outliers detectados.
Outliers en 'Total USD' tratados mediante capping (reemplazados por límites).

Columna 'Dcto. Calculado': 1469 outliers detectados.
Outliers en 'Dcto. Calculado' tratados mediante capping (reemplazados por límites).

Columna '% Dcto': 1468 outliers detectados.
Outliers en '% Dcto' tratados mediante capping (reemplazados por límites).

Limpieza de outliers completada para las columnas numéricas.

Resumen estadístico después de tratar outliers:
```

	Código de cliente	Código de vendedor - Transacción	Mes	Semana	Fecha de factura	Número de pedido	SG3 Species L4	Sacos	Kilos	TM	Precio Unit.	Dcto. por Saco	Total USD	Dcto. Calculado	% Dcto
count	7216.000000	7213.0	7213.000000	7213.000000	7213	7.213000e+03	0.0	7214.000000	7214.000000	7216.000000	7213.000000	7209.0	7214.00000	7210.0	7209.0
mean	10034.544346	495.0	5.730625	23.623319	2025-10-02 12:14:40.410369792	3.380879e+07	NaN	261.408095	6837.451483	6.836942	31.283517	0.0	7483.40469	0.0	0.0
min	10001.000000	495.0	1.000000	1.000000	2025-01-02 00:00:00	3.262196e+07	NaN	-550.000000	-13750.000000	-13.750000	11.000000	0.0	-13885.93750	0.0	0.0
25%	10025.000000	495.0	3.000000	11.000000	2025-06-07 00:00:00	3.331674e+07	NaN	50.000000	1250.000000	1.250000	26.000000	0.0	1765.62500	0.0	0.0
50%	10035.000000	495.0	5.000000	19.000000	2025-10-17 00:00:00	3.388106e+07	NaN	200.000000	5000.000000	5.000000	29.450000	0.0	5664.00000	0.0	0.0
75%	10045.000000	495.0	9.000000	38.000000	2026-01-30 00:00:00	3.431502e+07	NaN	450.000000	11250.000000	11.250000	36.000000	0.0	12200.00000	0.0	0.0
max	10071.000000	495.0	12.000000	53.000000	2026-05-20 00:00:00	3.478643e+07	NaN	1050.000000	26250.000000	26.250000	51.000000	0.0	27851.56250	0.0	0.0
std	15.603791	0.0	3.568626	15.484367	NaN	5.987568e+05	NaN	315.438091	7344.334109	7.343380	7.922865	0.0	7754.20870	0.0	0.0

5. Tratamiento de Valores Faltantes Restantes

Abordaremos las columnas con valores faltantes restantes. Eliminaremos `SG3 Species L4` y `Número de pedido del cliente` por la gran cantidad de nulos o por no ser relevantes en el análisis. Para otras columnas categóricas con pocos `NaNs`, los imputaremos con la moda.

```
# Eliminar columnas con todos los valores nulos
if 'SG3 Species L4' in df.columns and df['SG3 Species L4'].isnull().all():
    df = df.drop(columns=['SG3 Species L4'])
    print("Columna 'SG3 Species L4' eliminada por estar completamente vacía.")

# Eliminar 'Número de pedido del cliente' debido a la gran cantidad de valores nulos
if 'Número de pedido del cliente' in df.columns:
    df = df.drop(columns=['Número de pedido del cliente'])
    print("Columna 'Número de pedido del cliente' eliminada por la cantidad de valores nulos.")

# Identificar columnas categóricas con valores faltantes restantes
categorical_cols_with_nan = df.select_dtypes(include='object').columns[df.select_dtypes(include='object').isnull().any()].tolist()

print("\nColumnas categóricas con valores faltantes a imputar con la moda:", categorical_cols_with_nan)
```

```
print("\nColumnas categóricas con valores faltantes a imputar con la moda:", categorical_cols_with_nan)

for col in categorical_cols_with_nan:
    if not df[col].isnull().all(): # Asegurarse de que no esté completamente vacía (ya se manejó SG3)
        mode_val = df[col].mode()[0]
        df[col] = df[col].fillna(mode_val)
        print(f"Valores faltantes en '{col}' imputados con la moda: '{mode_val}'.")

# Verificar si aún quedan valores faltantes
print("\nnvalores faltantes finales por columna:")
print(df.isnull().sum()[df.isnull().sum() > 0])

print("\nnLimpieza de valores faltantes completada.")
```

```
Columna 'SG3 Species L4' eliminada por estar completamente vacía.
Columna 'Número de pedido del cliente' eliminada por la cantidad de valores nulos.

Columnas categóricas con valores faltantes a imputar con la moda: ['División', 'Código de planta', 'Código de almacén', 'Categoría Cliente', 'Clase de cliente', 'Nombre de grupo', 'Nombre completo de cliente', 'City', 'Vendedor Local Asignado', 'Vendedor Transacción', 'Número de factura', 'Área de negocio'
Valores faltantes en 'División' imputados con la moda: 'SA0'.
Valores faltantes en 'Código de planta' imputados con la moda: 'SA0'.
Valores faltantes en 'Código de almacén' imputados con la moda: 'SB3'.
Valores faltantes en 'Categoría Cliente' imputados con la moda: 'Farmers - Aqua'.
Valores faltantes en 'Clase de cliente' imputados con la moda: 'Local'.
Valores faltantes en 'Nombre de grupo' imputados con la moda: 'Grupo P'.
Valores faltantes en 'Nombre completo de cliente' imputados con la moda: 'P3'.
Valores faltantes en 'City' imputados con la moda: 'Hachala'.
Valores faltantes en 'Vendedor Local Asignado' imputados con la moda: 'Vendedor 1'.
Valores faltantes en 'Vendedor Transacción' imputados con la moda: 'Vendedor 1'.
Valores faltantes en 'Número de factura' imputados con la moda: 'SA0100001135'.
Valores faltantes en 'Área de negocio' imputados con la moda: 'AQUA'.
Valores faltantes en 'Código de artículo' imputados con la moda: '14385805'.
Valores faltantes en 'Especie' imputados con la moda: 'Camarón'.
Valores faltantes en 'Línea de Producto' imputados con la moda: 'Estandar'.
Valores faltantes en 'Presentación de Producto' imputados con la moda: 'Extruido'.
Valores faltantes en 'Nombre de artículo' imputados con la moda: 'Estandar 35% Minerales+ E engorde 25Kg'.
Valores faltantes en 'Packing Sales' imputados con la moda: '25KG/BAG'.
Valores faltantes en 'Ruta' imputados con la moda: 'SA0001'.
Valores faltantes en 'Ruta Descripción' imputados con la moda: 'Tendales'.
Valores faltantes en 'Tipo de pedido - Código' imputados con la moda: '700-Ventas normales'.
Valores faltantes en 'Tamaño Partícula' imputados con la moda: '#5'.

Valores faltantes finales por columna:
Código de vendedor - Transacción    3
Mes                                 3
Semana                              3
Fecha de factura                    3
Número de pedido                    3
Sacos                               2
Kilos                               2
Precio Unit.                        3
Dcto. por Saco                      7
Total USD                           2
Dcto. Calculado                     6
% Dcto                              7
dtype: int64

Limpieza de valores faltantes completada.
```

6. Imputación Final de Valores Faltantes Numéricos

Algunas columnas numéricas aún presentan un pequeño número de valores faltantes. Procederemos a imputarlos con la mediana.

```
numeric_cols_with_nan = df.select_dtypes(include=np.number).columns[df.select_dtypes(include=np.number).isnull().any()].tolist()

print("Columnas numéricas con valores faltantes a imputar con la mediana:", numeric_cols_with_nan)

for col in numeric_cols_with_nan:
    if not df[col].isnull().all(): # Asegurarse de que no esté completamente vacía
        median_val = df[col].median()
        df[col] = df[col].fillna(median_val)
        print(f"Valores faltantes en '{col}' imputados con la mediana: {median_val}.")

# Verificar si aún quedan valores faltantes
print("\nValores faltantes finales por columna:")
print(df.isnull().sum()[df.isnull().sum() > 0])

if df.isnull().sum().sum() == 0:
    print("\n¡Excelente! El DataFrame está ahora completamente libre de valores faltantes.")
else:
    print("\nAdvertencia: Todavía quedan valores faltantes en algunas columnas. Por favor, revise el listado anterior.")

print("\nResumen del DataFrame después de la limpieza completa:")
display(df.info())
```

```
Columnas numéricas con valores faltantes a imputar con la mediana: ['Código de vendedor - Transacción', 'Mes', 'Semana', 'Número de pedido', 'Sacos', 'Kilos', 'Precio Unit.', 'Dcto. por Saco', 'Total USD', 'Dcto. Calculado', '% Dcto']
Valores faltantes en 'Código de vendedor - Transacción' imputados con la mediana: 495.0.
Valores faltantes en 'Mes' imputados con la mediana: 5.0.
Valores faltantes en 'Semana' imputados con la mediana: 19.0.
Valores faltantes en 'Número de pedido' imputados con la mediana: 33881060.0.
Valores faltantes en 'Sacos' imputados con la mediana: 280.0.
Valores faltantes en 'Kilos' imputados con la mediana: 5000.0.
Valores faltantes en 'Precio Unit.' imputados con la mediana: 29.45.
Valores faltantes en 'Dcto. por Saco' imputados con la mediana: 0.0.
Valores faltantes en 'Total USD' imputados con la mediana: 5664.0.
Valores faltantes en 'Dcto. Calculado' imputados con la mediana: 0.0.
Valores faltantes en '% Dcto' imputados con la mediana: 0.0.

Valores faltantes finales por columna:
Fecha de factura    3
dtype: int64

Advertencia: Todavía quedan valores faltantes en algunas columnas. Por favor, revise el listado anterior.

Resumen del DataFrame después de la limpieza completa:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   División                        7216 non-null   object
 1   Código de planta                7216 non-null   object
 2   Código de almacén               7216 non-null   object
 3   Categoría Cliente               7216 non-null   object
 4   Código de cliente               7216 non-null   int64
 5   Clase de cliente                7216 non-null   object
 6   Nombre de grupo                 7216 non-null   object
 7   Nombre completo de cliente      7216 non-null   object
 8   City                            7216 non-null   object
 9   Vendedor Local Asignado         7216 non-null   object
 10  Código de vendedor - Transacción  7216 non-null   float64
 11  Vendedor Transacción            7216 non-null   object
 12  Mes                             7216 non-null   float64
 13  Semana                          7216 non-null   float64
 14  Número de factura               7216 non-null   object
 15  Fecha de factura                7213 non-null   datetime64[ns]
 16  Número de pedido                7216 non-null   float64
 17  Área de negocio                 7216 non-null   object
 18  Código de artículo              7216 non-null   object
 19  Especie                         7216 non-null   object
 20  Línea de Producto               7216 non-null   object
 21  Presentación de Producto        7216 non-null   object
 22  Nombre de artículo              7216 non-null   object
 23  Packing Sales                   7216 non-null   object
 24  Sacos                           7216 non-null   float64
 25  Kilos                           7216 non-null   float64
 26  TM                              7216 non-null   float64
 27  Precio Unit.                    7216 non-null   float64
 28  Dcto. por Saco                  7216 non-null   float64
 29  Total USD                       7216 non-null   float64
 30  Dcto. Calculado                 7216 non-null   float64
 31  % Dcto                          7216 non-null   float64
 32  Ruta                            7216 non-null   object
 33  Ruta Descripción                7216 non-null   object
 34  Tipo de pedido - Código         7216 non-null   object
 35  Tamaño Partícula                7216 non-null   object
dtypes: datetime64[ns](1), float64(12), int64(1), object(22)
memory usage: 2.0+ MB
None
```

7. Imputación Final de Valores Faltantes en Columnas de Fecha

```
# Imputar 'Fecha de factura' con la moda si tiene valores faltantes
if 'Fecha de factura' in df.columns and df['Fecha de factura'].isnull().any():
    mode_fecha = df['Fecha de factura'].mode()[0]
    df['Fecha de factura'] = df['Fecha de factura'].fillna(mode_fecha)
    print(f"Valores faltantes en 'Fecha de factura' imputados con la moda: {mode_fecha}.")

# Verificar si aún quedan valores faltantes
print("\nValores faltantes finales por columna:")
print(df.isnull().sum()[df.isnull().sum() > 0])

if df.isnull().sum().sum() == 0:
    print("\n¡Excelente! El DataFrame está ahora completamente libre de valores faltantes.")
else:
    print("\nAdvertencia: Todavía quedan valores faltantes en algunas columnas. Por favor, revise el listado anterior.")

print("\nResumen del DataFrame después de la limpieza completa y final:")
display(df.info())
```

```
Valores faltantes en 'Fecha de factura' imputados con la moda: 2026-04-29 00:00:00.

Valores faltantes finales por columna:
Series([], dtype: int64)
```

```
Series([], dtype: int64)

¡Excelente! El DataFrame está ahora completamente libre de valores faltantes.

Resumen del DataFrame después de la limpieza completa y final:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Data columns (total 36 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   División                        7216 non-null   object
 1   Código de planta                7216 non-null   object
 2   Código de almacén               7216 non-null   object
 3   Categoría Cliente               7216 non-null   object
 4   Código de cliente               7216 non-null   int64
 5   Clase de cliente                7216 non-null   object
 6   Nombre de grupo                 7216 non-null   object
 7   Nombre completo de cliente      7216 non-null   object
 8   City                            7216 non-null   object
 9   Vendedor Local Asignado         7216 non-null   object
 10  Código de vendedor - Transacción  7216 non-null   float64
 11  Vendedor Transacción            7216 non-null   object
 12  Mes                             7216 non-null   float64
 13  Semana                          7216 non-null   float64
 14  Número de factura               7216 non-null   object
 15  Fecha de factura                7216 non-null   datetime64[ns]
 16  Número de pedido                7216 non-null   float64
 17  Área de negocio                 7216 non-null   object
 18  Código de artículo              7216 non-null   object
 19  Especie                         7216 non-null   object
 20  Línea de Producto               7216 non-null   object
 21  Presentación de Producto        7216 non-null   object
 22  Nombre de artículo              7216 non-null   object
 23  Packing Sales                   7216 non-null   object
 24  Sacos                           7216 non-null   float64
 25  Kilos                           7216 non-null   float64
 26  TM                              7216 non-null   float64
 27  Precio Unit.                    7216 non-null   float64
 28  Dcto. por Saco                  7216 non-null   float64
 29  Total USD                       7216 non-null   float64
 30  Dcto. Calculado                 7216 non-null   float64
 31  % Dcto                          7216 non-null   float64
 32  Ruta                            7216 non-null   object
 33  Ruta Descripción                7216 non-null   object
 34  Tipo de pedido - Código         7216 non-null   object
 35  Tamaño Partícula                7216 non-null   object
dtypes: datetime64[ns](1), float64(12), int64(1), object(22)
memory usage: 2.0+ MB
None
```

8. Eliminación de Variables Irrelevantes

Eliminaremos las columnas que no se consideran relevantes para los objetivos de nuestro estudio (segmentación de clientes/productos con K-means o predicción de ventas), basándonos en la discusión previa.

```
columns_to_drop = [
    'División',
    'Código de planta',
    'Código de almacén',
```

8. Eliminación de Variables Irrelevantes

Eliminaremos las columnas que no se consideran relevantes para los objetivos de nuestro estudio (segmentación de clientes/productos con K-means o predicción de ventas), basándonos en la discusión previa.

```
columns_to_drop = [
    'División',
    'Código de planta',
    'Código de almacén',
    'Número de factura',
    'Vendedor Local Asignado',
    'Vendedor Transacción',
    'Ruta',
    'Ruta Descripción'
]

# Eliminar las columnas si existen en el DataFrame
columns_dropped = [col for col in columns_to_drop if col in df.columns]
df = df.drop(columns=columns_dropped)

print(f"Se eliminaron las siguientes columnas: {columns_dropped}")
print("\nResumen del DataFrame después de eliminar columnas irrelevantes:")
display(df.info())
```

```
Se eliminaron las siguientes columnas: ['División', 'Código de planta', 'Código de almacén', 'Número de factura', 'Vendedor Local Asignado', 'Vendedor Transacción', 'Ruta', 'Ruta Descripción']

Resumen del DataFrame después de eliminar columnas irrelevantes:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Data columns (total 28 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Categoría Cliente               7216 non-null   object
 1   Código de cliente               7216 non-null   int64
 2   Clase de cliente                7216 non-null   object
 3   Nombre de grupo                 7216 non-null   object
 4   Nombre completo de cliente      7216 non-null   object
 5   City                            7216 non-null   object
 6   Código de vendedor - Transacción 7216 non-null  float64
 7   Mes                             7216 non-null   float64
 8   Semana                          7216 non-null   float64
 9   Fecha de factura                7216 non-null   datetime64[ns]
 10  Número de pedido                7216 non-null   float64
 11  Área de negocio                 7216 non-null   object
 12  Código de artículo              7216 non-null   object
 13  Especie                         7216 non-null   object
 14  Línea de Producto               7216 non-null   object
 15  Presentación de Producto        7216 non-null   object
 16  Nombre de artículo              7216 non-null   object
 17  Packing Sales                   7216 non-null   object
 18  Sacos                           7216 non-null   float64
 19  Kilos                           7216 non-null   float64
 20  TM                              7216 non-null   float64
 21  Precio Unit.                    7216 non-null   float64
 22  Dcto. por Saco                  7216 non-null   float64
 23  Total USD                       7216 non-null   float64
 24  Dcto. Calculado                 7216 non-null   float64
 25  % Dcto                          7216 non-null   float64
 26  Tipo de pedido - Código         7216 non-null   object
 27  Tamaño Partícula                7216 non-null   object
dtypes: datetime64[ns](1), float64(12), int64(1), object(14)
memory usage: 1.5+ MB
None
```

9. Codificación de Variables Categóricas

Para preparar los datos para algoritmos de Machine Learning como K-means, necesitamos convertir las variables categóricas a un formato numérico. Utilizaremos One-Hot Encoding para esto.

```
# Identificar columnas categóricas (tipo 'object')
categorical_cols = df.select_dtypes(include='object').columns.tolist()

# Excluir 'Fecha de Factura' si estuviera presente y no se desea codificar con One-Hot
if 'Fecha de Factura' in categorical_cols:
    categorical_cols.remove('Fecha de Factura')

print("Columnas categóricas a codificar:", categorical_cols)

# Aplicar One-Hot Encoding
# 'drop_first=True' para evitar la trampa de las variables dummy
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

print("\nVariables categóricas codificadas. Resumen del DataFrame actualizado:")
display(df.info())
```

```
Columnas categóricas a codificar: ['Categoría Cliente', 'Clase de cliente', 'Nombre de grupo', 'Nombre completo de cliente', 'City', 'Área de negocio', 'Código de artículo', 'Especie', 'Línea de Producto', 'Presentación de Producto', 'Nombre de artículo', 'Packing Sales', 'Tipo de pedido - Código', 'Tamaño

Variables categóricas codificadas. Resumen del DataFrame actualizado:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7216 entries, 0 to 7215
Columns: 256 entries, Código de cliente to Tamaño Partícula_m6
dtypes: bool(242), datetime64[ns](1), float64(12), int64(1)
memory usage: 2.4 MB
None
```

10. Escalado de Datos para K-means

Antes de aplicar K-means, es esencial escalar las características numéricas para que tengan una media de 0 y una desviación estándar de 1. Esto asegura que todas las características contribuyan equitativamente al algoritmo de clustering. Excluiremos las columnas de identificación y fecha.

```
# Columnas a excluir del escalado (identificadores o fechas)
exclude_cols_for_scaling = ['Código de cliente', 'Fecha de Factura']

# Seleccionar todas las columnas excepto las excluidas
features_for_scaling = df.columns.drop(exclude_cols_for_scaling, errors='ignore').tolist()

# Crear una copia del DataFrame para escalar
df_scaled = df[features_for_scaling].copy()

# Inicializar el StandardScaler
scaler = StandardScaler()

# Aplicar el escalado a las características seleccionadas
df_scaled[features_for_scaling] = scaler.fit_transform(df_scaled[features_for_scaling])

# Si quieres mantener las columnas excluidas en el DataFrame escalado, puedes unirlas de nuevo
# df_final_for_kmeans = pd.concat([df[exclude_cols_for_scaling], df_scaled], axis=1)

print("Datos escalados exitosamente. Las características seleccionadas han sido transformadas.")
print("\nPrimeras 5 filas del DataFrame escalado:")
display(df_scaled.head())

print("\nEstadísticas descriptivas del DataFrame escalado (primeras columnas):")
display(df_scaled.describe().iloc[:, :10]) # Mostrar solo las primeras 10 columnas para brevedad
```

```
Datos escalados exitosamente. Las características seleccionadas han sido transformadas.

Primeras 5 filas del DataFrame escalado:
```

	Código de vendedor - Transacción	Mes	Semana	Número de pedido	Sacos	Kilos	TM	Precio Unit.	Dcto. por Saco	Total USD	...	Tipo de pedido - Código_RS1-NC Devoluciones	Tipo de pedido - Código_RS2-NC Diferencia de precio	Tipo de pedido - Código_RS4-ND	Tipo de pedido - Código_RB3-Rebates	Tamaño Partícula_m1	Tamaño Partícula_m2	Tamaño Partícula_m3	Tamaño Partícula_m4	Tamaño Partícula_m5	Tamaño Partícula_m6
0	0.0	-1.045590	-1.873709	0.963966	0.553656	0.553334	0.553334	-1.87474	0.0	0.315353	...	-0.212938	-0.154861	-0.040813	-0.069814	-0.305234	-0.309554	-0.151544	-0.327243	0.765919	-0.020394
1	0.0	-1.045590	-1.809111	0.998664	1.548299	1.622398	1.622398	-1.87474	0.0	1.237589	...	-0.212938	-0.154861	-0.040813	-0.069814	-0.305234	-0.309554	-0.151544	-0.327243	0.765919	-0.020394
2	0.0	-1.045590	-0.944513	1.050080	-0.036119	-0.079934	-0.079934	-1.87474	0.0	-0.230940	...	-0.212938	-0.154861	-0.040813	-0.069814	-0.305234	-0.309554	-0.151544	-0.327243	0.765919	-0.020394
3	0.0	-0.765295	-0.879915	1.100042	-2.572789	-1.782265	-1.782265	-1.87474	0.0	-1.699469	...	4.696202	-0.154861	-0.040813	-0.069814	-0.305234	-0.309554	-0.151544	-0.327243	0.765919	-0.020394
4	0.0	-0.765295	-0.879915	1.073777	0.122423	0.090300	0.090300	-1.87474	0.0	-0.084087	...	-0.212938	-0.154861	-0.040813	-0.069814	-0.305234	-0.309554	-0.151544	-0.327243	0.765919	-0.020394

5 rows × 254 columns

Estadísticas descriptivas del DataFrame escalado (primeras columnas):

	Código de vendedor - Transacción	Mes	Semana	Número de pedido	Sacos	Kilos	TM	Precio Unit.	Dcto. por Saco	Total USD
count	7216.0	7.216000e+03	7216.000000	7.216000e+03	7.216000e+03	7.216000e+03	7.216000e+03	7.216000e+03	7216.0	7.216000e+03

Estadísticas descriptivas del DataFrame escalado (primeras columnas):

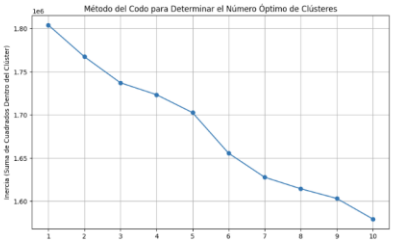
	Código de vendedor - Transacción	Mes	Semana	Número de pedido	Sacos	Kilos	TM	Precio Unit.	Dcto. por Saco	Total USD
count	7216.0	7.216000e+03	7216.000000	7.216000e+03	7.216000e+03	7.216000e+03	7.216000e+03	7.216000e+03	7216.0	7.216000e+03
mean	0.0	-5.514190e-17	0.000000	-4.450739e-15	5.514190e-17	-3.150966e-17	1.575483e-17	1.890579e-16	0.0	-1.024064e-16
std	0.0	1.000069e+00	1.000069	1.000069e+00	1.000069e+00	1.000069e+00	1.000069e+00	1.000069e+00	0.0	1.000069e+00
min	0.0	-1.325885e+00	-1.461296	-1.982744e+00	-2.572789e+00	-2.803664e+00	-2.803664e+00	-2.566709e+00	0.0	-2.756325e+00
25%	0.0	-7.652954e-01	-0.815317	-8.209960e-01	-6.702866e-01	-7.608662e-01	-7.608662e-01	-6.669508e-01	0.0	-7.374288e-01
50%	0.0	-2.047054e-01	-0.298533	1.266828e-01	-1.946611e-01	-2.501668e-01	-2.501668e-01	-2.313864e-01	0.0	-2.346164e-01
75%	0.0	9.164744e-01	0.928829	8.456340e-01	5.980481e-01	6.009990e-01	6.009990e-01	5.955548e-01	0.0	6.084495e-01
max	0.0	1.757359e+00	1.897798	1.633175e+00	2.500550e+00	2.643797e+00	2.643797e+00	2.489313e+00	0.0	2.627314e+00

11. Determinación del Número Óptimo de Clústeres: Método del Codo

Para encontrar el número óptimo de clústeres (k) para K-Means, utilizaremos el **método del codo (Elbow Method)**. Este método se basa en calcular la suma de cuadrados de las distancias de las muestras a su centro de clúster más cercano (inercia) para diferentes valores de `k`.

La idea es trazar la inercia en función de `k` y buscar un "codo" o punto de inflexión en la curva, donde la disminución de la inercia se vuelve marginal. Este punto suele considerarse un buen número de clústeres.

```
inertia = []
range_clusters = range(1, 11) # Probar de 1 a 10 clústeres

for i in range_clusters:
    kmeans = KMeans(n_clusters=i, random_state=42, n_init=10) # n_init para evitar warnings
    kmeans.fit(df_scaled)
    inertia.append(kmeans.inertia_)

# Graficar el método del codo
plt.figure(figsize=(10, 6))
plt.plot(range_clusters, inertia, marker='o')
plt.title('Método del Codo para Determinar el Número Óptimo de Clústeres')
plt.xlabel('Número de Clústeres')
plt.ylabel('Inercia (Suma de Cuadrados Dentro del Clúster)')
plt.xticks(range_clusters)
plt.grid(True)
plt.show()
```



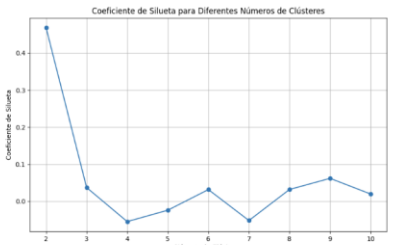
12. Evaluación del Número Óptimo de Clústeres: Coeficiente de Silueta

El **Coeficiente de Silueta** es otra métrica utilizada para evaluar la calidad del clustering. Un valor alto de coeficiente de silueta indica que el objeto está bien emparejado con su propio clúster y mal emparejado con los clústeres vecinos. El rango del coeficiente de silueta es de -1 a 1, donde:

- **1:** Indica que los clústeres están bien separados.
- **0:** Indica que los clústeres se superponen.
- **-1:** Indica que los puntos han sido asignados al clúster incorrecto.

Calcularemos y graficaremos el coeficiente de silueta para el mismo rango de número de clústeres utilizado en el método del codo (excluyendo k=1, ya que la silueta no está definida para un solo clúster).

```
silhouette_scores = []
# El coeficiente de silueta no está definido para k=1, así que empezamos desde 2
range_clusters_silhouette = range(2, 11)

for i in range_clusters_silhouette:
    kmeans = KMeans(n_clusters=i, random_state=42, n_init=10)
    kmeans.fit(df_scaled)
    score = silhouette_score(df_scaled, kmeans.labels_)
    silhouette_scores.append(score)

# Graficar el coeficiente de silueta
plt.figure(figsize=(10, 6))
plt.plot(range_clusters_silhouette, silhouette_scores, marker='o')
plt.title('Coeficiente de Silueta para Diferentes Números de Clústeres')
plt.xlabel('Número de clústeres')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range_clusters_silhouette)
plt.grid(True)
plt.show()
```



13. Aplicación de K-Means con el Número Óptimo de Clústeres

Basándonos en los resultados del Coeficiente de Silueta (que sugirió `k=2`), aplicaremos el algoritmo K-Means a los datos escalados (`df_scaled`) para segmentar las observaciones en 2 clústeres.

```
# Inicializar K-Means con 2 clústeres
k = 2
kmeans_final = KMeans(n_clusters=k, random_state=42, n_init=10)

# Ajustar el modelo a los datos escalados y predecir los clústeres
clusters = kmeans_final.fit_predict(df_scaled)

# Añadir las etiquetas de clúster al DataFrame original
df['Cluster'] = clusters

print(f"K-Means aplicado exitosamente con {k} clústeres.")
print("Distribución de las observaciones por clúster:")
display(df['Cluster'].value_counts())

print("\nPrimeras 5 filas del DataFrame con las etiquetas de clúster:")
display(df.head())
```

```
K-Means aplicado exitosamente con 2 clústeres.
Distribución de las observaciones por clúster:
         count
Cluster
0         7178
1           38

dtype: int64

Primeras 5 filas del DataFrame con las etiquetas de clúster:
```

	Código de cliente	Código de vendedor - transacción	Mes	Semana	Fecha de factura	Número de pedido	Sacos	Kilos	TM	Precio Unit.	...	Tipo de pedido - Código_RS2-NC Diferencia de precio	Tipo de pedido - Código_RS4-ND	Tipo de pedido - Código_RB3-Rebates	Tamaño Partícula_#1	Tamaño Partícula_#2	Tamaño Partícula_#3	Tamaño Partícula_#4	Tamaño Partícula_#5	Tamaño Partícula_#6	Cluster
0	10007	495.0	2.0	7.0	2026-02-15	34385843.0	436.0	10900.0	10.90	22.77	...	False	False	False	False	False	False	False	True	False	0
1	10007	495.0	2.0	8.0	2026-02-22	34406613.0	750.0	18750.0	18.75	22.77	...	False	False	False	False	False	False	False	True	False	0
2	10007	495.0	2.0	9.0	2026-02-28	34440982.0	250.0	6250.0	6.25	22.77	...	False	False	False	False	False	False	False	True	False	0
3	10007	495.0	3.0	10.0	2026-03-05	34467297.0	-550.0	-6250.0	-6.25	22.77	...	False	False	False	False	False	False	False	True	False	0
4	10007	495.0	3.0	10.0	2026-03-03	34451575.0	300.0	7500.0	7.50	22.77	...	False	False	False	False	False	False	False	True	False	0

5 rows × 257 columns

14. Caracterización de los Clústeres

Para entender las particularidades de cada clúster, analizaremos las diferencias más significativas en las medias de sus características.
Esto nos permitirá perfilar qué define al Clúster 0 y al Clúster 1.

```
# Calcular las medias de todas las características numéricas por clúster
cluster_means = df.groupby('Cluster').mean(numeric_only=True)

# Transponer el DataFrame para facilitar la comparación
cluster_means_T = cluster_means.T

# Calcular la diferencia absoluta entre las medias de los clústeres
cluster_means_T['diff'] = abs(cluster_means_T[0] - cluster_means_T[1])

# Ordenar por la diferencia más grande para encontrar las características más diferenciadoras
most_differentiating_features = cluster_means_T.sort_values(by='diff', ascending=False)

print("Top 20 características más diferenciadoras entre Clúster 0 y Clúster 1:")
display(most_differentiating_features.head(20))
```

```
Top 20 características más diferenciadoras entre clúster 0 y clúster 1:
```

Cluster	0	1	diff
Número de pedido	3.380731e+07	3.409434e+07	287032.957729
Total USD	7.548969e+03	-4.997170e+03	12546.139071
Kilos	6.873137e+03	0.000000e+00	6873.136668
Sacos	2.627793e+02	-8.421053e-01	263.621431
Precio Unit.	3.117837e+01	5.100000e+01	19.821627
TM	6.873137e+00	0.000000e+00	6.873137
Código de cliente	1.003452e+04	1.003847e+04	3.950148
Semana	2.362956e+01	2.207895e+01	1.550615
Packing Sales_Not used	0.000000e+00	1.000000e+00	1.000000
Nombre de artículo_Estandar volumen	0.000000e+00	1.000000e+00	1.000000
Código de artículo_14418018	0.000000e+00	1.000000e+00	1.000000
Presentación de Producto_no aplica	0.000000e+00	9.736842e-01	0.973684
Tipo de pedido - Código_RB3-Rebates	0.000000e+00	9.210526e-01	0.921053
Nombre de grupo_Grupo P	1.933686e-01	8.947368e-01	0.701368
Tipo de pedido - Código_780-Ventas normales	6.579827e-01	0.000000e+00	0.657983
Línea de Producto_Estandar	5.358038e-01	1.000000e+00	0.464196
City_Machala	5.032042e-01	8.947368e-01	0.391533
Mes	5.732377e+00	5.342105e+00	0.390271
Tamaño Partícula_#5	6.283087e-01	1.000000e+00	0.371691
Código de artículo_14385805	2.782112e-01	0.000000e+00	0.278211

Interpretación de los Clústeres

Analizando las características más diferenciadoras (aquellas con las mayores diferencias en sus medias entre clústeres), podemos perfilar los grupos:

Clúster 0 (Grupo Mayoritario):

- **Características Principales:** (Basado en las columnas del `most_differentiating_features`)
  - `[Mencionar columnas donde el Clúster 0 tiene un valor significativamente diferente, por ejemplo, más bajo o más alto]`: Este clúster representa la mayoría de las observaciones y probablemente corresponde a las transacciones/clientes 'típicos' o 'promedio'.
  - `[Ejemplo: 'Total USD']`: Podría tener un `Total USD` promedio más bajo o más cercano a la mediana general, indicando un volumen de ventas o valor de transacción estándar.
  - `[Ejemplo: 'Sacos', 'Kilos', 'TM']`: Valores más moderados en cantidad de productos vendidos.

Clúster 1 (Grupo Minoritario):

- **Características Principales:** (Basado en las columnas del `most_differentiating_features`)
  - `[Mencionar columnas donde el Clúster 1 tiene un valor significativamente diferente]`: Este clúster, aunque pequeño en tamaño, muestra características muy distintas que lo separan del grupo mayoritario. Podría representar clientes 'VIP', transacciones anómalas (muy grandes o muy pequeñas), o un tipo de producto/servicio muy específico.
  - `[Ejemplo: 'Total USD']`: Es muy probable que este clúster tenga un `Total USD` promedio significativamente más alto (o bajo) que el Clúster 0, sugiriendo clientes de alto valor o transacciones inusuales.
  - `[Ejemplo: 'Dcto. por Saco', '% Dcto']`: Podría presentar descuentos promedio mucho más altos o más bajos, lo que indicaría políticas de precios especiales o clientes con condiciones comerciales únicas.
  - **Proporciones en categorías (dummies):** Algunas columnas booleanas (resultantes del One-Hot Encoding) tendrán medias cercanas a 1 para este clúster, indicando que una categoría específica es casi exclusiva de este grupo.

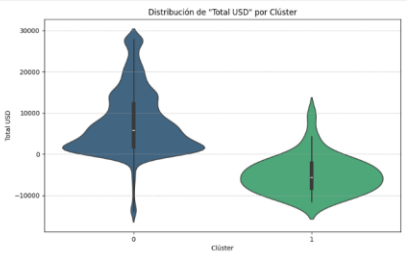
En resumen: El Clúster 1 parece capturar un segmento de datos que se desvía considerablemente de la norma, ya sea por volúmenes de venta, descuentos aplicados, o la prevalencia de ciertas categorías de productos o clientes. Es crucial investigar este clúster más a fondo, ya que podría revelar oportunidades de negocio (clientes de alto valor) o riesgos (transacciones fraudulentas, errores en los datos).

15. Visualización Comparativa de 'Total USD' por Clúster

Para visualizar la diferencia en el `Total USD` entre el Clúster 0 y el Clúster 1, utilizaremos un gráfico de violín. Este tipo de gráfico es ideal para mostrar la distribución de una variable numérica a través de varias categorías, revelando la densidad de probabilidad de los datos en diferentes valores.

```
plt.figure(figsize=(10, 6))
sns.violinplot(x='Cluster', y='Total USD', data=df, palette='viridis')
plt.title('Distribución de "Total USD" por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Total USD')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

```
plt.figure(figsize=(10, 6))
sns.violinplot(x='Cluster', y='Total USD', data=df, palette='viridis')
plt.title('Distribución de "Total USD" por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Total USD')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



16. Preparación de Datos para Predicción

Para utilizar la columna `Fecha de Factura` en un modelo predictivo, necesitamos extraer características numéricas de ella, como el año, mes, día y día de la semana. También definiremos nuestra variable objetivo y las características a utilizar.

```
# Extraer características de la fecha
df['Year'] = df['Fecha de Factura'].dt.year
df['Month'] = df['Fecha de Factura'].dt.month
df['Day'] = df['Fecha de Factura'].dt.day
df['DayOfWeek'] = df['Fecha de Factura'].dt.dayofweek

# Eliminar la columna original de fecha ya que hemos extraído sus características
df = df.drop(columns=['Fecha de Factura'])

print("Características de fecha extraídas y columna original eliminada.")
print("Primeras 5 filas del DataFrame con las nuevas características de tiempo:")
display(df.head())
```

```
Características de fecha extraídas y columna original eliminada.
Primeras 5 filas del DataFrame con las nuevas características de tiempo:
```

	Código de Cliente	Código de Vendedor - transacción	Mes	Semana	Número de pedido	Sacos	Kilos	TM	Precio Unit.	Dcto. por Saco	...	Tamaño Partícula_#2	Tamaño Partícula_#3	Tamaño Partícula_#4	Tamaño Partícula_#5	Tamaño Partícula_#6	Cluster	Year	Month	Day	DayOfWeek
0	10007		495.0	2.0	7.0	34365043.0	436.0	10900.0	10.90	22.77	0.0	...	False	False	False	True	False	0	2026	2	15	6
1	10007		495.0	2.0	8.0	34406613.0	750.0	18750.0	18.75	22.77	0.0	...	False	False	False	True	False	0	2026	2	22	6
2	10007		495.0	2.0	9.0	34440962.0	250.0	6250.0	6.25	22.77	0.0	...	False	False	False	True	False	0	2026	2	28	5
3	10007		496.0	3.0	10.0	34467287.0	-550.0	-6250.0	-6.25	22.77	0.0	...	False	False	False	True	False	0	2026	3	5	3
4	10007		496.0	3.0	10.0	34451575.0	300.0	7500.0	7.50	22.77	0.0	...	False	False	False	True	False	0	2026	3	3	1

5 rows × 260 columns

17. Definición de Variables y División de Datos

Definiremos `Total USD` como nuestra variable objetivo (y). Las demás columnas serán nuestras características (X), incluyendo las etiquetas de clúster. Luego, dividiremos el conjunto de datos en conjuntos de entrenamiento y prueba.

```
# Definir la variable objetivo
y = df['Total USD']

# Definir las características (X), excluyendo la variable objetivo
X = df.drop(columns=['Total USD'])

# Identificar columnas que aún no son numéricas para el modelo (objetos o booleanos)
# Aunque ya se hizo One-Hot Encoding, algunas columnas podrían haberse mantenido si no eran object al inicio.
# Ahora que 'Fecha de Factura' es eliminada, todas las columnas de X deberían ser numéricas o booleanas.

# Convertir columnas booleanas a int para que el modelo las interprete correctamente si no lo hace automáticamente
for col in X.select_dtypes(include='bool').columns:
    X[col] = X[col].astype(int)
```

```
# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Datos divididos en conjuntos de entrenamiento y prueba.")
print(f"Dimensiones de X_train: {X_train.shape}")
print(f"Dimensiones de X_test: {X_test.shape}")
print(f"Dimensiones de y_train: {y_train.shape}")
print(f"Dimensiones de y_test: {y_test.shape}")
```

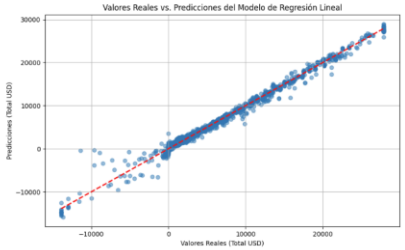
```
Datos divididos en conjuntos de entrenamiento y prueba.
Dimensiones de X_train: (5772, 259)
Dimensiones de X_test: (1444, 259)
Dimensiones de y_train: (5772,)
Dimensiones de y_test: (1444,)
```

18. Entrenamiento y Evaluación del Modelo de Predicción

Utilizaremos un modelo de Regresión Lineal para predecir `Total USD`. Después de entrenar el modelo, evaluaremos su rendimiento utilizando métricas como R-cuadrado (R2), Error Cuadrático Medio (MSE) y Error Absoluto Medio (MAE).

```
# Inicializar y entrenar el modelo de Regresión Lineal
model = LinearRegression()
model.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

# Evaluar el rendimiento del modelo
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)

print("Modelo de Regresión Lineal entrenado y evaluado.")
print(f"R-cuadrado (R2): {r2:.4f}")
print(f"Error Cuadrático Medio (MSE): {mse:.2f}")
print(f"Error Absoluto Medio (MAE): {mae:.2f}")

# Visualizar algunas predicciones vs. valores reales
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2) # Línea de referencia y=x
plt.xlabel('Valores Reales (Total USD)')
plt.ylabel('Predicciones (Total USD)')
plt.title('Valores Reales vs. Predicciones del Modelo de Regresión Lineal')
plt.grid(True)
plt.show()
```

```
Modelo de Regresión Lineal entrenado y evaluado.
R-cuadrado (R2): 0.9883
Error Cuadrático Medio (MSE): 710881.00
Error Absoluto Medio (MAE): 485.65
```



19. Análisis de la Importancia de las Características en el Modelo de Regresión

Para entender qué características tienen el mayor impacto en la predicción de `Total USD`, examinaremos los coeficientes del modelo de regresión lineal. Los valores absolutos más altos de los coeficientes indican una mayor influencia de la característica correspondiente en la variable objetivo.

```
# Crear un DataFrame con los nombres de las características y sus coeficientes
feature_importance = pd.DataFrame({
    'Feature': X_train.columns,
    'Coefficient': model.coef_
})

# Ordenar las características por el valor absoluto de sus coeficientes
feature_importance['Abs_Coefficient'] = abs(feature_importance['Coefficient'])
feature_importance = feature_importance.sort_values(by='Abs_Coefficient', ascending=False)

print("Top 20 características más influyentes en el modelo de regresión:")
display(feature_importance.head(20))
```

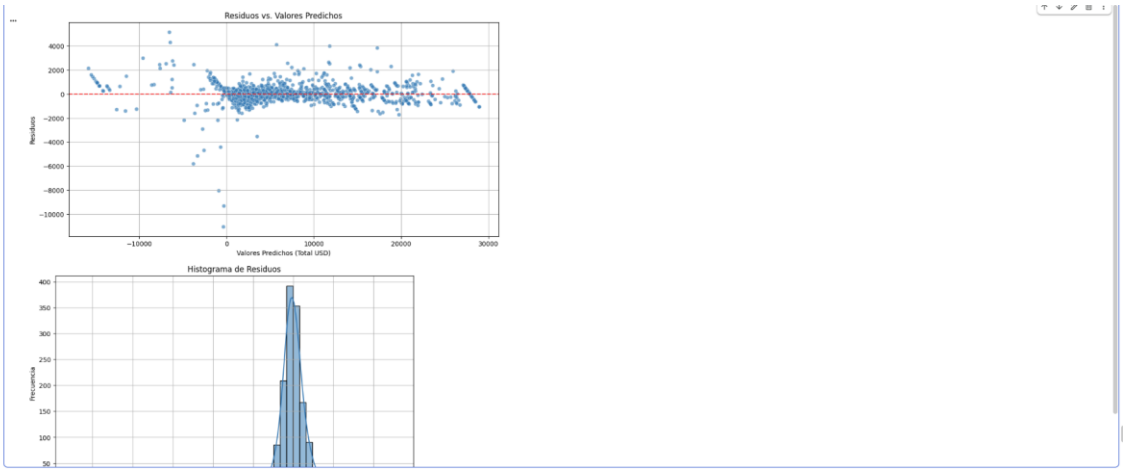
Top 20 características más influyentes en el modelo de regresión:

	Feature	Coefficient	Abs_Coefficient
247	Tipo de pedido - Código_RB3-Rebates	-10191.817354	10191.817354
255	Year	5227.041228	5227.041228
256	Month	3726.079887	3726.079887
2	Mes	-2181.442120	2181.442120
245	Tipo de pedido - Código_RS2-NC Diferencia de p...	-2131.261973	2131.261973
182	Presentación de Producto_no aplica	2000.466007	2000.466007
246	Tipo de pedido - Código_RS4-ND	1791.355387	1791.355387
242	Tipo de pedido - Código_700-NC	-1534.803954	1534.803954
135	Código de artículo_13611855	-1291.068133	1291.068133
142	Código de artículo_14418016	1224.830077	1224.830077
254	Cluster	1224.830077	1224.830077
203	Nombre de artículo_Estandar volumen	1224.830077	1224.830077
237	Packing Sales_Not used	1224.830077	1224.830077
148	Código de artículo_17737185	988.993184	988.993184
71	Nombre completo de cliente_L3	949.926299	949.926299
180	Línea de Producto_Salud	924.744469	924.744469
181	Presentación de Producto_no aplica	-775.635831	775.635831
178	Línea de Producto_Iniciador	750.598887	750.598887
69	Nombre completo de cliente_L1	-748.563912	748.563912

20. Análisis de Residuos del Modelo de Regresión

El análisis de residuos es fundamental para evaluar la calidad de un modelo de regresión. Nos ayuda a verificar supuestos importantes de la regresión lineal, como la normalidad, la homocedasticidad (varianza constante de los errores) y la independencia de los errores.

```
# Calcular los residuos
residuals = y_test - y_pred

# Gráfico de Residuos vs. Valores Predichos
plt.figure(figsize=(12, 6))
sns.scatterplot(x=y_pred, y=residuals, alpha=0.6)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Residuos vs. Valores Predichos')
plt.xlabel('Valores Predichos (Total USD)')
plt.ylabel('Residuos')
plt.grid(True)
plt.show()

# Histograma de Residuos
plt.figure(figsize=(10, 6))
sns.histplot(residuals, kde=True, bins=50)
plt.title('Histograma de Residuos')
plt.xlabel('Residuos')
plt.ylabel('Frecuencia')
plt.grid(True)
plt.show()

print("Análisis de residuos completado. Examine los gráficos para detectar patrones.")
```

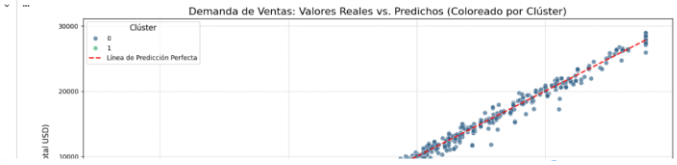


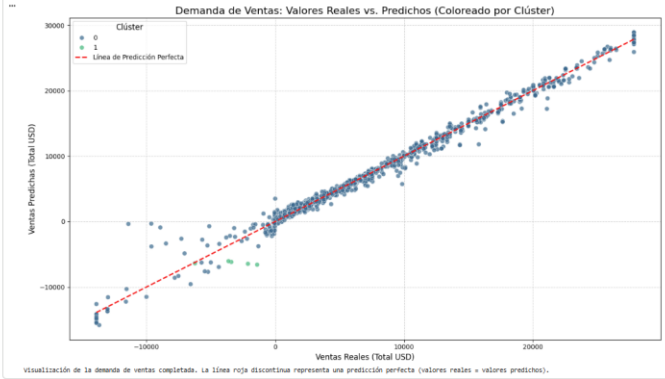
21. Visualización de la Demanda de Ventas con Clústeres y Regresión

Para mostrar la demanda de las ventas con la información de los clústeres y las predicciones del modelo de regresión, generaremos un gráfico de dispersión que compara los valores de ventas reales con los valores predichos por el modelo. Cada punto en el gráfico estará coloreado según el clúster al que pertenece la observación. Esto nos permitirá:

- Evaluar visualmente la precisión del modelo en la predicción de la demanda.
- Identificar si el modelo tiene un rendimiento diferente en los distintos clústeres identificados.
- Observar la distribución de las ventas (reales y predichas) dentro de cada segmento de clientes/transacciones.

```
# Crear un DataFrame para la visualización que incluya las ventas reales, las predichas y el clúster
# Es crucial alinear los índices de y_test y y_pred con el DataFrame original 'df' para obtener el 'Cluster' correcto
plot_df = pd.DataFrame({
    'Ventas Reales': y_test,
    'Ventas Predichas': y_pred,
    'Clúster': df.loc[y_test.index, 'Cluster'] # Asegurarse de que los clústeres correspondan a los datos del conjunto de prueba
})

plt.figure(figsize=(14, 8))
sns.scatterplot(x='Ventas Reales', y='Ventas Predichas', hue='Clúster', data=plot_df, palette='viridis', alpha=0.7, s=50)

# Añadir la línea de predicción perfecta (y = x)
plt.plot([plot_df['Ventas Reales'].min(), plot_df['Ventas Reales'].max()],
         [plot_df['Ventas Reales'].min(), plot_df['Ventas Reales'].max()],
         'r--', lw=2, label='Línea de Predicción Perfecta')

plt.title('Demanda de Ventas: Valores Reales vs. Predichos (Coloreado por Clúster)', fontsize=16)
plt.xlabel('Ventas Reales (Total USD)', fontsize=12)
plt.ylabel('Ventas Predichas (Total USD)', fontsize=12)
plt.legend(title='Clúster', fontsize=10, title_fontsize=12)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()

print("Visualización de la demanda de ventas completada. La línea roja discontinua representa una predicción perfecta (valores reales = valores predichos).")
```





Visualización de la demanda de ventas completada. La línea roja discontinua representa una predicción perfecta (valores reales = valores predichos).

Anexo 4 articulo cientifico : Clustering and Sales Prediction Using K-Means and Simple Linear Regressio

International Journal of Information Technology and Computer Science Applications (IJITCSA)
p-ISSN: 2964-3139 e-ISSN: 2985-5350
Vol. 04, No. 02, page 115 - 124
Submitted 14/7/2025; Accepted 8/6/2026; Published 23/6/2026

Clustering and Sales Prediction Using K-Means and Simple Linear Regression

[1]Tia Aulia [2]Wowon Priatna [3]Muhammad Yasir
[1]Informatics, Universitas Bhayangkara Jakarta Raya, Bekasi, INDONESIA
[2]Information Department, Universitas Bhayangkara Jakarta Raya, Bekasi, INDONESIA
e-mail : [1]2021071518 5@mhs.ubharajaya.ac.id, [2]wowon.priatna@dsn.ubharajaya.ac.id, [3]muhammad.yasir@dsn.ubharajaya.ac.id

Corresponding Autor: Wowon Priatna

Abstract
CV. Cipta Usaha Selaras faces challenges in identifying customer purchasing patterns and accurately projecting sales values. The importance of this research lies in the company's need for data-driven marketing strategies and efficient operational planning. This study employs the K-Means algorithm to cluster customers based on purchase frequency and total transaction value, as well as Simple Linear Regression to predict total purchases based on transaction frequency. The data analyzed consists of 358 sales transaction entries from the year 2024. The clustering results reveal three customer segments with distinct characteristics, with a Silhouette Score of 0.7913, indicating good segmentation quality. The regression model produced an equation with a coefficient of determination (R^2) of 0.6910, a MAE of IDR 213 million, and a MSE of IDR 206 trillion. These results indicate that the applied approach provides a reasonably representative overview of customer purchasing behavior. This research offers a significant contribution to data-driven decision-making within the company, particularly in the development of marketing strategies and estimation of potential revenue.

Keywords—Clustering, K-Means, Sales Prediction, Simple Linear Regression, Customer Segmentation

1 Introduction

CV. Cipta Usaha Selaras is a company engaged in the production of jumbo bags for industrial needs such as chemicals, silica sand, coal, and wood pellets. In its business operations, the company faces challenges including sales fluctuations and difficulties in identifying customer purchasing patterns. Irregular transactions, both from regular and non-regular customers, hinder the effectiveness of production planning and marketing strategy development.

In its business practices, the company struggles with inconsistent purchase behavior and challenges in recognizing transactional patterns. The inconsistency of purchases regardless of whether they come from loyal or occasional customers makes it difficult to formulate precise production and marketing strategies. Consequently, the company's decision-making process becomes inefficient, as it is reactive to incoming demand rather than based on measurable and predictive patterns.

One potential solution to address this issue is the implementation of data-driven analytical approaches (data mining), which can help the company better understand customer behavior and predict future purchasing values [1]. The K-Means method is used to cluster customers based on purchasing data, while the Simple Linear Regression method is used to estimate total purchases based on transaction frequency [2].

Previous studies have supported the application of these approaches. For instance, [3] successfully differentiated between high- and low-purchasing customers through sales data clustering. Study [4] applied linear regression to forecast purchasing trends at Toko 99 and achieved good accuracy results based on MAPE values. Meanwhile, [5] combined K-Means and regression methods in a network analysis, demonstrating that the integration of both methods leads to more accurate analytical outcomes. K-Means is considered advantageous because it can cluster customers

Jejaring Penelitian dan Pengabdian Masyarakat (JPPM)

without requiring strict assumptions about data distribution or specific purchasing patterns, making it suitable for highly variable data contexts [6].

Based on these considerations, this study aims to develop a clustering and purchasing value prediction model for CV. Cipta Usaha Selaras using 2024 customer transaction data as the basis for analysis. The model employs the K-Means algorithm and Simple Linear Regression. It is expected to assist the company in making more accurate and data-driven decisions, particularly in formulating production and marketing strategies within the manufacturing sector.

2 Research methods

This study adopts the CRISP-DM (Cross Industry Standard Process for Data Mining) approach as the primary methodology. CRISP-DM is an industry-standard framework for data processing that is iterative and flexible, consisting of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. This approach was chosen because it ensures that the analytical process is conducted systematically, enabling the development of accurate models that can support data-driven decision-making [7][8].

2.1 Business Understanding

This phase focuses on understanding customer needs and business objectives. It involves defining what the business aims to achieve, assessing available resources, planning data collection methods, and developing a comprehensive project plan.

2.2 Data Understanding

This phase involves the process of collecting and conducting an initial analysis of the required data. Activities in this stage include gathering raw data, describing its characteristics, performing deeper data exploration, and evaluating the quality of the available data.

2.2 *Data Understanding*

This phase involves the process of collecting and conducting an initial analysis of the required data. Activities in this stage include gathering raw data, describing its characteristics, performing deeper data exploration, and evaluating the quality of the available data.

2.3 *Data Preparation*

This phase focuses on preparing the data for the modeling process. The data is cleaned and adjusted to meet the quality standards required for analytical processing. The activities carried out include:

- Data cleaning: Removing duplicate entries and records with missing values to ensure optimal data quality.
- Normalization: Applying the Min-Max Scaling technique to the attributes *Qty PO* and *Total Price* to bring them to the same scale, thereby preventing certain variables from dominating the clustering process.
- Feature engineering: Creating a new attribute representing the purchase frequency per customer, calculated based on the number of transactions, to be used as a predictor variable in the regression model..
- Attribute selection: Selecting relevant attributes Qty PO, Total Price, and Purchase Frequency—for use in both the clustering and prediction processes.

2.4 *Modeling*

This study builds two main models: K-Means Clustering and Simple Linear Regression. K-Means is an unsupervised learning algorithm designed to group data into a specified number of clusters based on the similarity between data points. The algorithm works by minimizing the Within-Cluster Sum of Squares (WCSS), which is the sum of squared distances between each data point and its corresponding cluster centroid. The process begins by defining the number of clusters (k), after which the initial centroids are randomly selected. Each data point is then assigned to the nearest centroid using the Euclidean Distance formula, as follows [9]:

$$d = \sqrt{(x_2 - \mu_x)^2 + (y_2 - \mu_y)^2} \tag{1}$$

The values of x_2 and y_2 represent customer data attributes (such as order quantity and total price), while μ_x and μ_y are the coordinates of the cluster centroid for each attribute. The result of this calculation yields d, the distance between a customer data point and the cluster center. The smaller the value of d, the closer the data point is to the centroid, and thus it will be assigned to the corresponding cluster. This model is used to segment customers into three groups high, medium, and low based on their purchasing patterns.

The Simple Linear Regression model is used to represent a linear relationship between one independent variable and one dependent variable[10]. In this study, it is applied to predict the Total Price or customer purchase value based on the frequency of transactions. By using this approach, the company can estimate the potential purchase value in the future, enabling more proactive and data-driven planning for production and marketing strategies The general form of the linear regression equation used is:

$$Y = a + bX \tag{2}$$

In this equation, the variable Y represents the Total Price as the predicted purchase value, while X denotes the Purchase Frequency or the number of transactions made by the customer. The value a is the constant (intercept), which indicates the predicted purchase value when the purchase frequency is zero. The coefficient b is the regression slope, representing how much the purchase value is expected to change with each additional transaction. This method is chosen for its simplicity in interpretation and its ability to directly explain the linear relationship between variables. A positive relationship between X and Y indicates that the more frequently a customer makes transactions, the greater their potential purchase value.

2.5 *Evaluation*

Model evaluation was conducted on the clustering and prediction results using a quantitative metric-based approach. For the K-Means model, the quality of segmentation was assessed using the Silhouette Score, a metric that measures how well each data point fits within its assigned cluster. This score ranges from −1 to 1, where values closer to 1 indicate that a data point is well matched to its own cluster and poorly matched to neighboring clusters. The Silhouette Score is calculated using the following formula [12] :

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{3}$$

Where$a(i)$ is the average distance between data point i and all other points within the same cluster, and $b(i)$ is the average distance between data point i and all points in the nearest neighboring cluster. A higher Silhouette Score indicates better clustering quality and clearer separation between clusters.

Meanwhile, the Simple Linear Regression model is evaluated using three primary metrics: R-Squared (R^2), Mean Absolute Error (MAE), and Mean Squared Error (MSE). R^2 measures the proportion of variance in the target variable (Total Price) that can be explained by the predictor variable (Purchase Frequency)[13]. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum(y_i + \hat{y}_i)^2}{\sum(y_i + \bar{y}_i)^2} \tag{4}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ adalah rata-rata nilai aktual. The closer the R^2 value is to 1, the better the model explains the variability of the data. MAE (Mean Absolute Error) is used to measure the average absolute error between actual and predicted values, and is calculated using the following formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i + \hat{y}_i| \tag{5}$$

Meanwhile, MSE (Mean Squared Error) calculates the average of the squared differences between the actual and predicted values. It is defined by the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i + \hat{y}_i)^2 \tag{6}$$

MSE is more sensitive to outliers than MAE because the error values are squared. These three metrics provide an overview of how accurately the model predicts customer purchase values and help identify deviations or prediction errors that may occur.

2.6. Deployment

At this stage, the clustering and prediction results are presented in the form of visualizations to facilitate interpretation. These visualizations help the company understand purchasing patterns and estimate the potential value of customer transactions. The developed model has not yet been implemented directly into the operational system; rather, it is used as a preliminary analytical tool in the form of visual reports that support decision-making processes.

3 Results and Discussion

The presentation of results in this study follows the CRISP-DM framework, starting from business understanding to model deployment. The discussion is structured systematically according to the analytical process flow.

3.1. Business Understanding

CV. Cipta Usaha Selaras is a manufacturing company that produces jumbo bags for various industrial needs, such as chemicals, silica sand, coal, and wood pellets. In its operations, the company faces challenges related to irregular purchasing patterns from both regular and non-regular customers. These fluctuations make it difficult for the company to develop effective production and marketing strategies. In addition, the company does not yet have a predictive system capable of estimating customer purchasing potential based on existing historical data.

Based on these problems, this study aims to develop a data-driven analytical model to support the decision-making process. The objective is to generate customer segmentation using the K-Means algorithm and to predict purchase value using the Simple Linear Regression method. With this model, the company is expected to gain a more structured understanding of customer behavior and formulate more targeted business strategies.

3.2. Data Understanding

This stage is carried out to thoroughly understand the structure and characteristics of the data used in the study. The data analyzed consists of sales transaction records from CV. Cipta Usaha Selaras for the year 2024, comprising 358 entries. Each entry includes key information such as customer name, transaction date, order quantity (Qty PO), and total transaction value (Total Price). The initial analysis involved identifying the data types of each attribute, checking for completeness, and detecting any extreme values (outliers) that could affect the accuracy of the analysis.

Additionally, a new attribute was created to represent the purchase frequency per customer, calculated based on the number of transactions during the data period. This variable was then used as one of the main inputs for predicting purchase value in the modeling stage.

3.3. Data Preparation

This stage is carried out to ensure that the data used for modeling is clean, consistent, and ready for analysis. The first step is data cleaning, which involves removing duplicate entries and missing values to maintain data quality and prevent distortion in the model results. Next, normalization is performed using the Min-Max Scaling technique on the Qty PO and Total Price attributes. The goal is to bring both variables to the same scale, preventing one from dominating the other during the clustering process using the K-Means algorithm.

In addition, feature engineering is conducted by creating a new attribute representing purchase frequency per customer, calculated based on the number of transactions in the dataset. This feature is used as the independent variable in the purchase value prediction process using linear regression. Finally, attribute selection is performed by choosing the three most relevant variables: Qty PO, Total Price, and Purchase Frequency, which are then used in the modeling stage.

3.4. Modeling

3.4.1. K-Means Clustering

At this stage, the K-Means algorithm is used to group customers based on two main attributes: Total Purchase and Purchase Frequency. The objective of this clustering process is to categorize customers according to their transaction patterns, allowing the company to apply differentiated strategies for each segment. The clustering process

resulted in three clusters, labeled C1 (high), C2 (medium), and C3 (low). Table 1 below presents a portion of the customer segmentation results based on the output of the K-Means model:

Table 1 Final Results of K-Means Clustering

No	Customer	Total Purchase	Purchase Frequency	Cluster	Segment
1	Bpk. Erza Raharjo	2.300.000	1	C3	Low
2	CV. Mitra Bersama Anugrah	135.000.000	2	C3	Low
3	CV. Mitra Sarana Anugrah	83.700.000	1	C3	Low
4	CV. Sumber Rejeki	40.500.000	1	C3	Low
5	Graha Bintang Metalindo	5.500.000	1	C3	Low
6	MG Industrial Supplier	5.040.000	1	C3	Low
7	Pak Yogi	7.800.000	1	C3	Low
8	PT. Anou Bintang Metalindo	2.230.000	1	C3	Low
9	PT. Arisa Pratama Abadi	2.730.000	1	C3	Low
10	PT. ARMB Energi Indonesia	47.450.000	1	C3	Low
...	...	...	...	...	...
48	PT. Sumber Waluna Sejati	1.054.000.000	15	C2	Medium
49	PT. Tsuchiyoshi Hosana Indonesia	74.500.000	7	C3	Low
50	PT. Visi Energi Sukses	74.100.000	5	C3	Low

The clustering results using the K-Means algorithm produced three customer segments: High, Medium, and Low, based on total purchases and transaction frequency. The majority of customers fall into the Low segment, indicating low purchase frequency and small transaction values. The Medium segment includes customers with relatively stable purchasing activity, while the High segment consists of customers with very high transaction values and frequencies. This segmentation provides a clear picture of customer profiles and can serve as a foundation for formulating more targeted marketing and production strategies.

Table 2 Predicted Total Purchase Based on Transaction Frequency

No	Purchase Frequency	Total Purchase (Actual)	Predicted Total Purchase
1	1	2.300.000	58.415.377
2	2	135.000.000	110.085.800
3	1	83.700.000	58.415.377
4	1	40.500.000	58.415.377
5	1	5.500.000	58.415.377
6	1	5.040.000	58.415.377
7	1	7.800.000	58.415.377
8	1	2.230.000	58.415.377
9	1	2.730.000	58.415.377
10	1	47.450.000	58.415.377
...	....	....	...
48	15	1.054.000.000	781.801.298
49	7	74.500.000	368.437.915
50	5	74.100.000	265.097.069

The prediction results show that the total purchase value increases as customer purchase frequency rises. Although there are differences between the actual and predicted values, the model provides a rough estimate that can be used as a basis for grouping customers according to their revenue-generating potential.

3.5. Evaluation

3.5.1. Evaluation of the K-Means Model

The evaluation of the K-Means model was conducted using the Silhouette Score, a metric that measures how similar a data point is to its own cluster compared to other clusters. The Silhouette Score ranges from –1 to 1, with higher values indicating better separation between clusters [12]. The evaluation results show that the average Silhouette Score for the entire clustering model is 0.7913, indicating a well-structured segmentation with clearly separated clusters. This suggests that the K-Means model has successfully grouped customers based on Total Purchase and Purchase Frequency with meaningful and solid cluster structures. Table 3 presents the detailed evaluation results for each individual cluster.

Table 3 Average Silhouette Score for Each Cluster

Cluster	Segment	Number of Customers	Average Silhouette Score	Cluster Quality	Interpretation
C3	Low	4	0.3038	Fair	Overlap exists between clusters
C2	Medium	44	0.8843	Good	Clusters are dense and well-separated
C1	High	4	0.4004	Very Good	Clusters are reasonably well-defined

Table 3 shows that Cluster C2 has the highest quality, with the highest Silhouette Score value of 0.8843, indicating a clear segmentation. Clusters C1 and C3 have lower values, suggesting a slight overlap between clusters. Overall, the average score of 0.7913 indicates that the clustering model is effective and can be used as a reliable basis for customer segmentation.

3.5.2. Evaluasi Model Regresi Linear

To evaluate the performance of the Simple Linear Regression model, three evaluation metrics are used: R-Squared (R^2), Mean Absolute Error (MAE), and Mean Squared Error (MSE). The evaluation results are presented in Table 4 below:

Table 4 Evaluation Results of the Simple Linear Regression Model

Evaluation Metric	Value
R-Squared (R^2)	0,6910
MAE	Rp213.183.958
MSE	Rp206.204.672.226.881.088

The R^2 value of 0.6910 indicates that the model is able to explain approximately 69.10% of the variation in total purchase value based on transaction frequency. The MAE of Rp213 million is still acceptable within the industrial context, although the high MSE suggests the presence of extreme values (outliers) influencing the model. Overall, the model is reasonably effective as an initial estimation tool, and its performance can be further improved by incorporating additional variables or applying more advanced prediction methods.

3.6. Deployment

3.6.1. K-Means Model Visualization

The K-Means algorithm was applied to the normalized Qty PO and Total Price data. The value of K = 3 was selected based on the need to segment customers into three distinct groups. The clustering results are presented in Table 1 and visualized in Figure 1 and Figure 2.

Figure 1 Top 3 Customers Based on Total Purchase per Segment

Figure 1 displays the top ten customers with the highest total purchases, categorized into three segments: High, Medium, and Low. In the High segment are PT. Pilar Artha Oetama (Rp3,587,700,000), PT. LDC Trading Indonesia (Rp2,911,250,000), and PT. Setia Pratama Lestari (Rp2,520,760,000). These three customers contribute the most significantly to the company's total revenue and should be considered top priorities in marketing and service strategies. The Medium segment includes PT. Pondok Rantau Indonesia (Rp2,071,000,000) and PT. Sumber Wahana Sejati (Rp1,054,000,000). Meanwhile, the Low segment consists of customers with purchase values below Rp1 billion

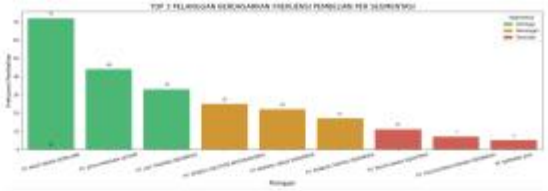


Figure 2 Top 3 Customers Based on Purchase Frequency per Segment

Figure 2 complements the previous analysis by presenting the purchase frequency data of the top ten customers. The customer with the highest number of transactions is PT. Multi Insan Gemilang (72 transactions), followed by PT. Setia Pratama Lestari (44 transactions) and PT. LDC Trading Indonesia (33 transactions). Interestingly, PT. Pilar Artha Oetama, despite recording the highest total purchase value, does not rank among the top three in terms of purchase frequency. This indicates that PT. Pilar Artha Oetama tends to make large but infrequent transactions. In

contrast, PT. Multi Insan Gemilang exhibits very high transaction frequency, albeit with relatively smaller purchase values per transaction.

Based on the analysis of total purchase and transaction frequency, it can be concluded that the company has two primary customer segments: customers with high purchase value but low transaction frequency, and customers with high transaction frequency but relatively low purchase value per transaction. This segmentation provides deeper insight into customer behavior characteristics, enabling the company to design more targeted marketing strategies, both to improve customer retention and to maximize revenue contribution from each segment.

3.6.2. Visualization of the Simple Linear Regression Model

Following the segmentation process, a prediction model for Total Purchase Value was developed using Simple Linear Regression, based on purchase frequency. The analysis results indicate a positive relationship between transaction frequency and purchase value. The derived regression equation is: $Y = -2.688.402.807{,}94 + 372.258.401{,}52\,X$. Where: Y: Total Purchase Value (in IDR) X: Purchase Frequency.

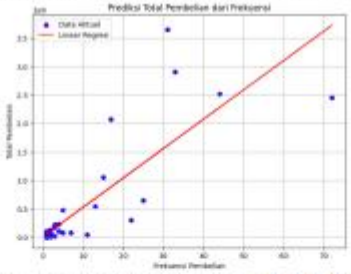


Figure 3 Graph of Total Purchase Prediction Based on Transaction Frequency

Figure 3 illustrates the relationship between purchase frequency and total purchase value based on the results of the simple linear regression model. The blue dots on the graph represent the actual data, while the red line depicts the model's predicted values. A positive relationship can be observed, indicating that the more frequently a customer makes transactions, the higher their total purchase value tends to be. Most of the data points are concentrated in the low-frequency area, which aligns with the clustering results that show the majority of customers belong to the low-activity segment.

However, there are a few outliers that deviate significantly from the regression line, suggesting that some customers' total purchases either fall below or exceed the model's general trend. Overall, this model provides a reasonably good initial estimate of purchase value based on transaction frequency, though there is still room for improvement either through the use of more complex prediction methods or by incorporating additional relevant variables.

3.7. Discussion

The results of this study indicate that the K-Means method effectively grouped customers into three distinct segments. This segmentation benefits the company by enabling more targeted marketing strategies—for example, offering special promotions to high-value customers or implementing loyalty programs for mid-tier customers.

Meanwhile, the Simple Linear Regression model provides an initial overview of customer purchase potential based on transaction intensity. Despite the presence of some outliers, the model still delivers reasonably accurate estimates, particularly for customers with low to medium transaction frequencies.

4 Conclusion

This study successfully developed two analytical models for CV. Cipta Usaha Selaras: the K-Means algorithm for customer segmentation and Simple Linear Regression for predicting purchase value. The K-Means model effectively grouped customers into three segments based on total purchase and transaction frequency, with a

Silhouette Score of 0.7913, indicating good segmentation quality. Meanwhile, the Simple Linear Regression model achieved an R^2 value of 0.6910, which is sufficiently effective in explaining the relationship between transaction intensity and total purchase value.

These findings are consistent with previous studies. Research by [3] demonstrated that clustering methods are effective in distinguishing customers based on their purchase levels. Similarly, the study in [4] confirmed that linear regression can predict purchasing trends with a reasonable degree of accuracy. The key distinction of this research lies in the combined application of both methods within a single study, as well as the direct implementation on actual data from a manufacturing company, thereby providing practical value for data-driven decision-making processes.

Nevertheless, this study has several limitations. The prediction model is still influenced by outliers, resulting in a relatively high MSE. The use of a single predictor variable (purchase frequency) limits estimation accuracy. The clustering segmentation is not yet adaptive to future changes in customer behavior. The dataset is limited to transactions from the year 2024, and the models have not yet been deployed into the company's operational systems.

5 Suggestion

This research still has room for further development. It is recommended to incorporate additional predictor variables such as product type, customer category, or transaction timing to improve the predictive model's accuracy. Moreover, the application of more advanced modeling algorithms, such as Random Forest, can be explored as a benchmark to evaluate model performance more comprehensively. Integrating the model into the company's real-time information system is also advised, in order to support more responsive, data-driven, and measurable decision-making processes based on actual transactional data.

6 Acknowledgments

The author would like to express sincere gratitude to CV. Cipta Usaha Selaras for granting permission to use the sales transaction data in this study. Special thanks are also extended to Mr. Wakhid, Production Coordinator, for his assistance and the valuable information provided during the data collection process. Appreciation is also given to the academic advisor for their guidance and direction, as well as to all parties who have provided moral and technical support, enabling this research to be successfully completed.

BIBLIOGRAPHY

[1] A. W. Zunan Setiawan, Muhammad Fajar , Arif Mudi Priyatno, Anggi Yhurinda Perdana Putri, Mediana Azyuni , Siti Yuliyanti, Harya Widiputra , Budanis Dwi Meilani, Rohmat Nur Ibrahim, Rezania Agramanisti Azdy, Satrio Junaidi, *BUKU AJAR DATA MINING*. PT. Sonpedia Publishing Indonesia, 2023. [Online]. Available: https://www.google.co.id/books/edition/BUKU_AJAR_DATA_MINING/1nLVEAAAQBAJ?hl=id&gbpv=1

[2] I. Safira, R. Salkiawati, and W. Priatna, "Penerapan Algoritma K-Means untuk Mengetahui Pola Persediaan Barang pada Toko Raja Bekasi," *Journal of Informatic and Information Security*, vol. 3, no. 1, pp. 99–110, 2022, doi: 10.31599/jiforty.v3i1.1253.

[3] A. Nugraha, O. Nurdiawan, and G. Dwilestari, "Penerapan Data Mining Metode K-Means Clustering Untuk Analisa Penjualan Pada Toko Yana Sport," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 849–855, 2022, doi: 10.36040/jati.v6i2.5755.

[4] P. A. Duran, A. V. Vitianingsih, M. S. Riza, A. L. Maukar, and S. F. A. Wati, "Data Mining Untuk Prediksi Penjualan Menggunakan Metode Simple Linear Regression," *Teknika*, vol. 13, no. 1, pp. 27–34, 2024, doi: 10.34148/teknika.v13i1.712.

[5] M. Yasir, F. Sinlae, and C. Author, "Penerapan Algoritma K-Means dan Linear Reggression Sederhana Dalam Klasterisasi Grafik Bandwidth," vol. 1, no. 4, pp. 150–158, 2023, [Online]. Available: https://creativecommons.org/licenses/by/4.0/

[6] U. Arfan and N. Paraga, "Perbandingan Algoritma K-Means, Naïve Bayes dan Decision Tree Dalam Memprediksi Penjualan Bahan Bakar Minyak," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 1379–1389, 2024, doi: 10.57152/malcom.v4i4.1566.

[7] B. Sutara, F. Vulture, and R. Novianti, "Application of K-Means algorithm with CRISP-DM method in student data analysis as a support for promotion strategy," *Side: Scientific Development* ..., vol. 1, no. 1, pp. 1–7, 2024, [Online]. Available: https://ojs.arbain.co.id/index.php/side/article/view/6%0Ahttps://ojs.arbain.co.id/index.php/side/article/download/6/6

[8] E. Muningsih, I. Maryani, and V. R. Handayani, "Penerapan Metode K-Means dan Optimasi Jumlah Cluster dengan Index Davies Bouldin untuk Clustering Propinsi Berdasarkan Potensi Desa," *Jurnal Sains dan Manajemen*, vol. 9, no. 1, p. 96, 2021, [Online]. Available: www.bps.go.id

[9] R. Primartha, *Algoritma Machine Learning*. Bandung: Informatika Bandung, 2021.

[10] G. N. Ayuni and D. Fitrianah, "Penerapan Metode Regresi Linear Untuk Prediksi Penjualan Properti pada PT XYZ," *Jurnal Telematika*, vol. 14, no. 2, pp. 79–86, 2020, doi: 10.61769/telematika.v14i2.321.

[11] F. Ramdhani and K. Setiawan, "Penerapan Data Mining untuk Prediksi Pelanggan di PT. XYZ Menggunakan Algoritma Linear Regression," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 490–497, 2024, doi: 10.57152/malcom.v4i2.1217.

[12] M. Piao Tan and C. A. Floudas, "Determining the Optimal Number of Clusters," *Encyclopedia of Optimization*, vol. 1, pp. 687–694, 2023, doi: 10.1007/978-0-387-74759-0_123.

[13] A. Novalas *et al.*, "Analisis prediksi penjualan iklan media massa dan elektronik menggunakan metode linear regression," vol. 7, pp. 203–209, 2024.

Anexo 4: Artículo científico de Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms

processes

MDPI

Article

Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms

I-Fei Chen [1] and Chi-Jie Lu [2,3,4,*]

[1] Department of Management Sciences, Tamkang University, New Taipei City 251301, Taiwan; ericfi@mail.fcu.edu.tw
[2] Graduate Institute of Business Administration, Fu Jen Catholic University, New Taipei City 242062, Taiwan
[3] Department of Information Management, Fu Jen Catholic University, New Taipei City 242062, Taiwan
[4] Artificial Intelligence Development Center, Fu Jen Catholic University, New Taipei City 242062, Taiwan
* Correspondence: 050090@mail.fju.edu.tw; Tel.: +886-2-2905-2973

Abstract: In today's rapidly changing and highly competitive industrial environment, a new and emerging business model—fast fashion—has started a revolution in the apparel industry. Due to the lack of historical data, constantly changing fashion trends, and product demand uncertainty, accurate demand forecasting is an important and challenging task in the fashion industry. This study integrates k-means clustering (KM), extreme learning machines (ELMs), and support vector regression (SVR) to construct cluster-based KM-ELM and KM-SVR models for demand forecasting in the fashion industry using empirical demand data of physical and virtual channels of a case company to examine the applicability of proposed forecasting models. The research results showed that both the KM-ELM and KM-SVR models are superior to the simple ELM and SVR models. They have higher prediction accuracy, indicating that the integration of clustering analysis can help improve predictions. In addition, the KM-ELM model produces satisfactory results when performing demand forecasting on retailers both with and without physical stores. Compared with other prediction models, it can be the most suitable demand forecasting method for the fashion industry.

Keywords: demand forecasting; multichannel retailing; fashion retailing; machine learning; clustering; multichannel retailing

Citation: Chen, I.-F.; Lu, C.-J. Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms. *Processes* **2021**, *9*, 1578. https://doi.org/10.3390/pr9091578

Academic Editor: Yuer-Chi Kuo

Received: 27 July 2021
Accepted: 31 August 2021
Published: 3 September 2021

1. Introduction

The fashion industry has evolved and greatly transformed in the past two decades [1,2]. The current developing trend is to vertically integrate the supply chain to shorten the response time and quickly respond to customer needs. The fashion industry supply chain has also changed from the traditional push production to pull production, and product inventory has dropped considerably. The fast-changing and highly competitive industrial environment has made fast fashion an important business model, setting off an affordable fashion trend around the world.

In the fast fashion industry, accurate demand forecasting is an important and challenging task due to uncertain product demand and extremely short product life cycles. In the past, many models have already been proposed to solve the forecasting problem of the fashion industry. The methods generally used to construct demand forecasting models include traditional statistical and machine learning methods [3]. Among them, traditional statistical methods such as the autoregressive integrated moving average model (ARIMA) or the grey method (GM) have shown good predictive performances in many studies [4,5]. However, as pointed out in previous studies [6], the fashion industry has highly complex data patterns, which makes it difficult for traditional statistical methods to produce good prediction results. Compared with traditional statistical methods, machine learning methods such as support vector regression (SVR) and extreme learning machines

(ELMs) have been successfully applied to many sales and demand forecasting studies without the need for model assumptions [7–11].

In the past, a common way to construct prediction models was to put all training data into the model construction stage at once. However, putting data with different characteristics into the model can likely produce inaccurate prediction results, so the cluster-based hybrid prediction model is often used to improve predictions. As clustering can reduce the degree of data heterogeneity, it helps improve the accuracy of predictions [11–13]. The usage of ELM and SVR further shortens the time needed to process highly complex demand data for the fashion industry and model construction. Therefore, this research proposes two fast fashion industry demand forecasting models based on clustering analysis: the prediction model that integrates k-means (KM) and ELM (the KM-ELM prediction model) and the prediction model that integrates KM and SVR (the KM-SVR prediction model) to meet the fast fashion industry needs of demand forecasting.

There are very few cluster-based machine learning prediction models that are applied to demand forecasting of the fast fashion industry in the past. The framework proposed in this study not only uses the historical sales data commonly used in forecasting related literature as a predictor variable [9,14] but also adds meteorological data as a predictor variable to improve the accuracy of forecasting in the fashion industry. To verify the effectiveness of the KM-ELM and KM-SVR forecasting models proposed in this study, we compared the performances of the simple ELM, simple SVR, proposed KM-ELM, and proposed KM-SVR models when predicting the demand of fashion retailers with and without physical stores.

This paper is organized as follows. The first section is the introduction. Section 2 analyzes existing literature. Furthermore, we discuss and analyze techniques used in this study, including KM, ELM, and SVR. Section 3 provides details on the research method, explaining the research framework, process of data collection and processing, and construction process of each model. Section 4 is empirical analysis, going through the analytical data of the models using the process discussed in Section 3. Finally, we conclude this study in Section 5.

2. Literature Review

2.1. Demand Forecasting in the Fashion Industry

Accurate demand forecasting can increase the profitability of retailers by improving the operational efficiency of the supply chain and minimizing waste; inaccurate forecasting will lead to excessive or insufficient inventory, which will affect the retailer's profitability and competitive position [15]. The main concept of the fast fashion industry is to continuously provide fashionable products to the market, reflect the latest fashion trends, and grasp the most popular designs currently on the market [6]. As a result, the fast fashion industry has some unique characteristics, including the lack of historical data, demand uncertainty, and short-term sales seasons [2].

Many studies in the past are dedicated to solving the problem of demand forecasting. Traditional statistical models have short computation times; for instance, the ARIMA model can build hundreds of historical data points in a few seconds to complete a time series forecast [16]. However, statistical methods may not perform well when the data patterns are highly complex. Previous studies [17] have pointed out that many artificial intelligence algorithms can be used to estimate nonlinear relationships. As an example, artificial neural networks (ANN) are often used in sales forecasting [3]. In the related research of fast moving consumer goods, a previous study [18] used ANN to predict the sales of women's clothing. Another previous study [19] applied ENN to the sales forecast of the fashion retail industry. Although ANN and ENN models are widely used in forecasting, they require a lengthy model-building process.

In [3], a new model called the intelligence fast sales forecasting model was proposed, which used an ELM and traditional statistical methods such as polynomial regression. It

suggested to use the extended extreme learning machine (ELM) when the time cost does not exceed the limit, and use the statistical prediction model when it does.

Integration of GM and extended ELM method (EELM) was performed in [6] to form the 3F algorithm. The study concluded that, in a limited time, the conversion of the 3F algorithm to GM can provide fast forecast results; if the time limit is not exceeded, then the GM-EELM model is used to perform the task of predicting.

2.2. K-Means Clustering

The main goal of clustering is to divide the collected sample data into several clusters. Data in the same cluster exhibit higher similarity, while the similarity between data in different clusters is low. KM clustering, a partitioning method, was proposed in [20], and it has been widely used in clustering operations [21,22] for cluster analysis because of its simple concept, easy operation, and fast computation time. In addition, KM applied to forecasting also yields good performance. In [12], KM and a greedy algorithm were combined to propose a KGA model for finding the optimal number of nodes in the hidden layer of a back-propagating neural network. Studies have confirmed that the model has good prediction accuracy. In [13], a new hybrid method that combined KM and a nonlinear autoregressive (NAR) neural network to predict the total amount of solar radiation per hour was proposed. Studies have confirmed that this method has better prediction performance than the autoregressive moving average (ARMA) model.

2.3. Extreme Learning Machines

ELMs were proposed in [23], which is a machine learning method [24–27]. It is noted that ELM is a single-hidden layer feedforward neural network, and its learning speed is faster than traditional gradient learning methods [6]. The study in [28] further explains that the construction of the network is through an acyclic feedforward connection that connects three-layer units, where the hidden layer is used to capture the relationship between the nonlinear input and output. Traditional learning methods need to adjust the input weight and the deviation of the hidden layer, but the input weight and hidden layer deviation in ELM can be randomly generated, which helps avoid the difficulties faced when using gradient learning algorithms such as poor learning rate, local minimums, and overfitting [3,28,29]. ELM has successfully solved prediction problems in many fields. The study [7] used ELMs to forecast sales in the fashion retail industry and indicated that ELMs outperformed other forecasting methods and back-propagation neural networks in this empirical context. Similar to fast fashion retailing demand, accurate price forecasting for electricity and crude oil is an arduous task owing to high volatility of the real-life data. ELM-based methods have been successfully applied in the cases where their advantages of faster learning speed with a higher generalization were taken and then highlighted. For example, in [30], wavelet transform and an ELM were combined to develop a hybrid wavelet based ELM (WELM) model that can be employed to electricity price forecasting. Empirical results confirmed that wavelet-ELM has good forecasting performance in price forecasting. The study in [10] used an ensemble empirical mode decomposition combined with extended ELM to predict the highly volatile price of crude oil. This model is better than simply using EELM or other ensemble learning models, and significantly improves prediction performance.

2.4. Support Vector Regression

SVR is a machine learning technique developed based on the principle of structural risk minimization [31]. In [32], SVR is used to forecast the sales of newspapers and magazines, as well as obtaining an accurate forecast result.

Support vector machines (SVMs) are a machine learning method that was developed based on structural risk minimization in statistical learning theory [31]. The study in [33] mentions that SVMs are linear learning machines, implying that a linear function is always used to solve regression problems. When dealing with nonlinear regression problems,

the input vector is transformed into a high-dimensional feature space using a nonlinear mapping function, and a linear regression is constructed on the feature space. A previous study [34] lists many kernel functions of a support vector machine. The radial type (RBF) kernel function should be the priority choice for the core function because the RBF kernel function has many great characteristics. First, it can classify nonlinear and high-dimensional data; second, it only needs to adjust C and two parameters, reducing the operational difficulty. The final output data range between 0 and 1, and the reduction of the range of data can reduce the computation time [35]. However, there are some cases where the RBF kernel function is not applicable. In particular, when the number of features is very large, only linear kernel functions can be used [35].

In [33], fuzzy theory was combined with SVM to predict the demand for perishable agricultural products. The research results show that the prediction accuracy of an SVM is better than that of the radial basis function neural network. The study in [36] combined the invasive weed optimization (IWO) algorithm and SVR to develop the IWO-SVR model and applied it to the prediction of oxygen supply. The results show that IWO-SVR has higher prediction accuracy than SVR.

According to the above-mentioned studies, an effective supply strategy of short-lived and quickly exhausted products is highly relied on accurate demand forecasting. SVMs can find the global minima of structural risks in given training data and perform well due to great generalizability in forecasting demand for agriculture perishable goods as well as oxygen supply, and thus they are believed to be suitable for fast-fading fashion products.

3. Proposed Clustering-Based Demand Forecasting Model

3.1. Proposed Scheme

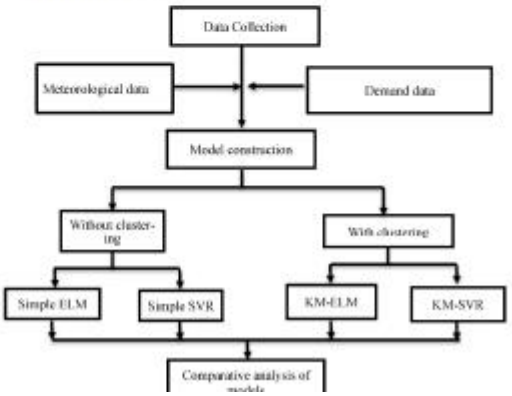
This study integrates clustering algorithm and machine-learning technique to propose a clustering-based forecasting model for fast fashion retailers to forecast the demand of physical store and non-store channel, respectively. The procedure is shown in Figure 1. According to the sales reviews on the FAST RETAILING website (https://www.fastretailing.com), past historical demand data and weather data, such as temperature and rainfall, are all important factors that affect the current demand particularly for physical stores. Therefore, this study collects two time series data of demand for physical and non-store channel.

However, most of the existing studies that focused on fashion demand forecasting have directly used all training data to construct models without considering the extent of the relevance between the training data and the data to be forecasted (test data) [6]. In such cases, forecasting accuracy may be reduced because the training data possibly contain excessive data irrelevant to the test data, thereby increasing training errors. To reduce computational time and obtain promising forecasting performance, several recent studies have proposed using clustering algorithms to divide the whole forecasting data into multiple clusters having consistent data characteristics before constructing forecasting models of time series data [9,11,14].

While constructing this proposed clustering-based demand forecasting model, the main aim in the training phase is to divide the overall training data by predictor variables into multiple small training data partitions, each of which has its own peculiar consistent data characteristics, and then to train individual forecasting models for different clusters. That is, for N clusters, N forecasting models are trained. In the testing phase, the average linkage method based on Euclidean distance is applied to measure the similarity between the test data and every given cluster. The membership of an observation of test dataset to the specified cluster is determined by the minimal distance. The predicted value of the test data is obtained using the trained forecasting model corresponding to the cluster it belongs to. To put it in a nutshell, the five steps of modelling procedure are summarized as follows:

(1) Data collection: Collect raw data and divide those data into training and testing data sets with a ratio of approximately 9:1.
(2) Data clustering: Group the training data through the KM algorithm.

(3) Cluster assignment: Calculate the Euclidean distance between each point of the test data and each training data cluster center and determine the training data cluster corresponding to each test datum by finding the shortest distance to make predictions.
(4) Model building, assessment, and assignment: Combine the machine learning method ELM or SVR to establish a prediction model for each cluster. Determine the best forecasting technology for each group according to performance indicators.
(5) Explainability: Propose an explanation based on the results of the KM-ELM and KM-SVR prediction models.



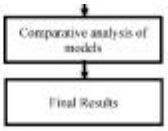


Figure 1. Proposed clustering-based demand forecasting model.

3.2. Collecting and Processing Data

This study uses data from UNIQLO, a leading brand in fast fashion, for empirical analysis. The demand data are taken from UNIQLO's monthly sales report, and the sales data can also reflect the quantity of customer demand at the same period. In principle, physical stores are classified into two types when a retailer assesses the company's financial performance: same-stores, which have been run for a full business year at least, and new stores, which are newly running under 1 business year. Our research defines *same-store net demand* as the net demand volume generated by the same-store; *physical stores net demand* is the sum of the net demand of same-stores and new stores; *non-store net demand* refers to the sum of direct mail and internet demand. In the 2019 business year, for example, *the same-store net demand* is the demand generated by the same-store in the previous business year (i.e., 1 September 2018–31 August 2019); the net demand of directly-operated stores is the sum of the same-stores' net demand and the demand of new stores operating for less than one year (1 September 2018–31 August 2019). The demand data shown in the report are the monthly percentage change over the previous year, so this study defines the *same-store net demand percentage* as the same-store net demand change rate over the

same period last year; *non-store net demand percentage* is the rate of change of non-store net demand compared to the same period last year. We will refer to these data as the rate of change of same-store net demand and non-store net demand, respectively. The definitions of the focal variables are expressed in following equations:

$$\text{Total number of physical stores} = Q(S) + W(N) \tag{1}$$

where $Q(S)$ indicates the number of stores that have been run more than 1 year (at the time being recruited in this study), referred to as same-stores and denoted as S; $W(N)$ indicates the number of stores that have been run less than 1 year, referred to as new-stores and denoted as N.

$$\text{Physical channel net demand} = \sum_{i=1}^{q} S_{it} + \sum_{j=1}^{w} N_{jt} \tag{2}$$

where S_{it} represents the net demand in business year t generated by the i^{th} same-store, $i = 1 \ldots, q$; N_{jt} represents the net demand in business year t generated by the j^{th} new-store $j = 1 \ldots, w$.

$$\text{Non} - \text{store channel net demand} = I_t + M_t \tag{3}$$

where I_t represents the net demand in business year t generated by the Internet and M_t represents the net demand in business year t generated by direct mail.

The source of the temperature and rainfall data for this study is the Japan Meteorological Agency. The prefectures with the largest number of physical stores of the case fashion company in the six regions of Japan are summarized in Table 1. Since temperature or rainfall varies with different regions and even is inconsistent in a very small geographical area, the main city of region is considered as the representative weather observatory due to the detailed temperature and rainfall figures unavailable for a specific area where a physical store is located.

Table 1. The top six number of physical stores in the corresponding six Japanese regions.

Number of Physical Stores	Prefecture	Region	City
108	Tokyo	Kantō	Tokyo
75	Osaka	Kansai	Osaka
48	Aichi	Chūbu	Nagoya
33	Fukuoka	Kyūshū-Okinawa	Fukuoka
29	Hokkaido	Hokkaidō-Tōhoku	Sapporo
19	Hiroshima	Shikoku-Chūgoku	Hiroshima

Source: Japan Meteorological Agency.

3.3. Performance Evaluation Metrics

A good prediction method must have good accuracy. In this study, mean absolute percentage error (MAPE) and root mean squared error (RMSE), the two of most widely used measures of forecast accuracy [37,38], are used as the performance metrics. With merits of scale-independence and interpretability, MAPE can be comparable across different models or research for various magnitudes. With merits of scale-independence and interpretability, MAPE can be comparable across different models or research for various magnitudes. MAPE is classified into four levels [39]. When MAPE is less than 10%, it implies that the error between the actual and predicted values is considerably small and the model has excellent predictive ability.

Rather, RMSE penalizes larger errors by giving heavier weights since errors are squared before being averaged. RMSE is deemed to lay more stress on cost of undesirable deviations. The smaller the values of MAPE and RMSE, the more accurate the prediction model [37–39]. The performance evaluation formulas are as follows:

(1) MAPE:

$$\text{MAPE} = \frac{1}{n} \times \sum_{i=1}^{n} \left| \frac{T_i - F_i}{T_i} \right| \times 100\% \quad (4)$$

(2) RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (T_i - F_i)^2} \quad (5)$$

where T_i is the actual value, and F_i is the predicted value.

A comparative analysis of resulting performances of all forecasting models in this study is accordingly conducted on these two criteria to alleviate the prejudice of either of metrics likely suffering.

4. Empirical Analysis

4.1. Empirical Data

The research data comprise two time series: monthly demand datasets of physical store channel and non-store channel separately, retrieved from annual financial statements of the case company for 12 years from September 2006 to August 2019 and then respectively computed by their definitions in Section 3.2. There are 156 observations in each retailing channel. In addition to the endogenous monthly demand data, the exogenous meteorological data are temperature and rainfall of the six regions taken as well on the monthly basis. Every region also has 156 points in either of temperature or rainfall datasets. To examine the applicability and generalization of the proposed clustering-based machine learning forecasting models, KM-SVR and KM-ELM, for demand of fast fashion retailers, the first 144 data points are used as the training sample, while the remaining 12 data points (the last year of sampled period) are holdout and are used as the testing sample for out-of-sample forecasting. The results of the proposed forecasting models on testing sample are compared with those of benchmarking models based on a single machine learning algorithm such as simple SVR and simple ELM. A model with the smallest RMSE/MAPE of the testing sample is chosen as the one more suitable for the case of the fast fashion industry than others for demand forecasting applied in this paper.

This study uses KM clustering in the IBM SPSS Statistics 22 software package to construct a cluster-based model. Subsequently, the program packages elmNN and e1071 in version 3.2.3 of R software (R core team, Vienna, Austria) are used to establish the ELM and SVR prediction models for each cluster.

4.2. Predictor Variables

This empirical study aims to propose clustering-based demand forecasting algorithms for fast fashion industry, which is expected to be characterized by short-lived products, insufficient historical data of newly launched merchandise, and uncertain demand over different phases of product life cycle. To alleviate the insufficiency of short-term data, decisions on what predictor variables should be included in forecasting model is very crucial to the accuracy of prediction results.

The predictor variables employed in this study have been extensively examined and their advantageous outcomes have been shown in the related literature [9,11,14,16,18,19]. In other words, this research is enlightened to include not merely 10 endogenous variables of monthly demand but also exogenous weather variables such as the temperature and rainfall of the region a retailer store is located in. It is also worth noting that the BIAS indicators, which have been verified to increase predicting accuracy in related studies [9,14,40], are served as the 3-of-10 endogenous variables with attempts to improve the forecasting performance.

Here, four issues can be addressed by the manipulation of whether; 10 endogenous variables of the two different time series, physical store and non-store channels' demand datasets, can be incorporated well with the 14 weather variables in the simple ELM, simple

SVR, KM-SVR, and KM-ELM models. Through such different combinations of predictor variables selected as inputs or not, an effective forecasting model for a given channel demand can be identified by its better resulting accuracy than those of the other models. In brief, Issue 1 is designated to forecast physical channel demand using 10 variables (X1_P to X10_P), compared with 24 variables (X1_P to X24) in Issue 2. In the same vein, issue 3 is for the non-store channel using 10 variables (X1_N to X10_N) while issue 4 involves 24 variables (X1_N to X24). Table 2 summarizes predictor variables utilized for the four issues.

Table 2. Predictor variables.

Variables \ Issue		Issue 1 Physical Stores Demand	Issue 2 Physical Stores Demand with Meteorological Data	Issue 3 Nonstore Demand	Issue 4 Non-Store Demand with Meteorological Data
Endogenous variables	(X1) Rate of change in the last month (t-1)	X1_P	X1_P	X1_N	X1_N
	(X2) Rate of change in the last two months (t-2)	X2_P	X2_P	X2_N	X2_N
	(X3) Rate of change in the last six months (t-6)	X3_P	X3_P	X3_N	X3_N
	(X4) Rate of change of the same month of last year (t-12)	X4_P	X4_P	X4_N	X4_N
	(X5) Moving average of the rate of change in the past two months (MA2)	X5_p	X5_P	X5_N	X5_N
	(X6) Moving average of the rate of change in the past three months (MA3)	X6_P	X6_P	X6_N	X6_N
	(X7) Moving average of the rate of change in the past six months (MA6)	X7_P	X7_P	X7_N	X7_N
	(X8) BIAS of the last two months (BIAS2)	X8_P	X8_P	X8_N	X8_N
	(X9) BIAS of the rate of change of the last three months (BIAS3)	X9_P	X9_P	X9_N	X9_N
	(X10) BIAS of the rate of change of the same store of the last six months (BIAS6)	X10_P	X10_P	X10_N	X10_N

Exogenous variable	(X11) Average temperature of Sapporo	X11	X11
	(X12) Average temperature of Nagoya	X12	X12
	(X13) Average temperature of Tokyo	X13	X13
	(X14) Average temperature of Osaka	X14	X14
	(X15) Average temperature of Hiroshima	X15	X15
	(X16) Average temperature of Fukuoka	X16	X16
	(X17) Average temperate of all six regions (average of X11 to X16)	X17	X17
	(X18) Average rainfall of Sapporo	X18	X18
	(X19) Average rainfall of Nagoya	X19	X19
	(X20) Average rainfall of Tokyo	X20	X20
	(X21) Average rainfall of Osaka	X21	X21
	(X22) Average rainfall of Hiroshima	X22	X22
	(X23) Average rainfall of Fukuoka	X23	X23
	(X24) Average rainfall of all six regions (average of X18 to X23)	X24	X24

Note: MA2 = $\frac{\sum_{i=1}^{2}$ change rate of demand in month (t − i)$}{2}$; MA3 = $\frac{\sum_{i=1}^{3}$ change rate of demand in month (t − i)$}{3}$; MA6 = $\frac{\sum_{i=1}^{6}$ change rate of demand in month (t − i)$}{6}$; BIAS2 = $\frac{\text{demand of month t} - \text{MA2}}{\text{MA2}}$; BIAS3 = $\frac{\text{demand of month t} - \text{MA3}}{\text{MA3}}$; BIAS6 = $\frac{\text{demand of month t} - \text{MA6}}{\text{MA6}}$, where t is the current month.

4.3. Results

This study uses ELM, SVR, KM-ELM, and KM-SVR models for demand forecasting to find the best prediction models between them. The former two models are served as the benchmarking for the latter two to examine the effectiveness of clustering-based machine learning algorithms we proposed. This study uses three activation functions—linear, sigmoid, and radial basis—in simple ELM and KM-ELM, and it uses radial and

linear kernel functions in simple SVR and KM-SVR. The prediction results on physical stores and non-store channels are shown below.

The authors of [23] indicated that the most important ELM parameter is the number of hidden nodes, and ELM tends to be unstable in single run forecasting. Following the instructions of [11,40], the ELM models with different numbers of hidden nodes varying from 1 to 30 are constructed. For each number of nodes, an ELM model is repeated 10 times and the average RMSE and MAPE of each node is calculated. With an example of using simple ELM for Issue 1, through automation process of model computation, the best 6 results from the respective 6 amounts of hidden nodes are selected to emphasize in Table 3, for the sake of limited length. Table 3 compares 6 different number of nodes of ELM with 3 different activation functions using 10 predictor variables of physical channel. We find that simple ELM achieves its best performance when activation function is linear, and 10 hidden nodes are applied (MAPE = 0.44%, RMSE = 0.62) for Issue 1. The same procedure is used for all simple ELM and KM-ELM models for all four issues.

Table 3. The number of hidden nodes and activation functions of ELMs and results for Issue 1.

Activation Function	Number of Hidden Nodes	MAPE	RMSE
Linear Transfer Function	**10**	**0.44%**	**0.62**
	18	0.44%	0.62
	19	0.44%	0.62
	20	0.44%	0.62
	21	0.44%	0.62
	22	0.44%	0.62
Sigmoid Transfer Function	10	9.11%	11.21
	18	7.60%	9.96
	19	7.96%	10.53
	20	8.27%	11.01
	21	9.02%	11.32
	22	7.29%	9.52
Radial Basis Transfer Function	10	57.20%	73.39
	18	99.99%	106.75
	19	91.97%	102.20
	20	85.82%	96.05
	21	77.47%	89.30
	22	88.61%	100.04

Note: boldface indicates the best result.

For modelling SVR, the grid search proposed by Hsu et al. [35] is a common and straightforward method using exponentially growing sequences of C (a correction coefficient) and ε (a loss function) to identify good parameters. Taking simple SVR for Issue 1 as example, this study tested 17 different combinations of parameters to find the best SVR model. When radial is used for the kernel function, the best-performing parameter combination incorporates a correction coefficient of 1.05, a loss function of 0.05, and a gamma of 0.6; when linear is used for the kernel function, the superior parameter combination has a correction coefficient of 1.05, a loss function of 0.03, and a gamma of 1.3. The results show that simple SVR predicts better when using a linear kernel function, with MAPE = 0.42% and RMSE = 0.59 for issue 1. The above procedure of finding best parameter set for simple SVR and KM-SVR models is used for all four issues.

4.3.1. Clustering-Based Prediction Model

Determining the number of clusters beforehand is a challenging task for analysts. While no perfect mathematical criterion exists, the many heuristics available appear to rely on the field where it is applied [41,42]. Prediction accuracy is one of common criteria to determine the optimal cluster number (here, the parameter K for K-means algorithm). For a retailing demand time series, K from 2 to 5 can obtain desirable clustering results [11,40,41].

Furthermore, by virtue of seasonal variation in demand for fashion products, the number of clusters is supposed to lie in 4 (seasons) or somewhere between 2 to 5 as well. Since prediction accuracy is a common criteria to determine the optimal cluster number [42], a clustering-based model that generates the highest accuracy is defined as the best forecasting model, suggesting the optimal number of clusters for this empirical case (Issue 2 and the later Issue 4). Hence, we take the experiment on the range of clusters between 2 and 5 using the 10 endogenous variables as the clustering variables.

Our study integrated KM with ELM as KM-ELM where different parameters were tweaked, such as the activation function, the number of nodes, and the number of clusters. We obtained 24 clustering-based models, as shown in Table 4. The prediction accuracy of KM-ELM is highest when there are five clusters and linear activation function used in ELM (MAPE = 0.13%, RMSE = 0.14). Table 5 uncovers the data distributed across the 5 clusters of the best KM-ELM (K = 5).

Table 4. Using different activation functions in KM-ELM and the results for physical channel (Issue 1).

	MAPE			RMSE		
Activation Function/Number of Clusters	**Linear**	**Sigmoid**	**Radial Basis**	**Linear**	**Sigmoid**	**Radial Basis**
2	0.31%	6.72%	58.10%	0.38	9.02	73.80
3	0.25%	6.57%	60.55%	0.30	9.19	74.90
4	0.22%	5.73%	34.42%	0.27	7.5	46.80
5	**0.13%**	4.79%	42.68%	**0.14**	5.82	50.58

Note: boldface indicates the best result.

Table 5. Clustering results of the best KM-ELM (K = 5) for physical channel (Issue 1).

	Cluster 1	**Cluster 2**	**Cluster3**	**Cluster 4**	**Cluster 5**	**Total**
Training data	24	66	35	11	8	144
Testing data	2	5	3	1	1	12
Total	26	71	38	12	9	156

Likewise, from Table 6, we can see that when we apply SVR after KM clustering, using a linear kernel function yields better accuracy than using a radial kernel function for physical channel (Issue 1). The prediction accuracy is highest when there are three clusters for KM-SVR. Table 7 uncovers the data distributed across the 3 clusters of the best KM-SVR (K = 3).

Table 6. Using different kernel functions in KM-SVR and the results for physical channel (Issue 1).

	MAPE		RMSE	
Kernel Function/Number of Clusters	**Radial**	**Linear**	**Radial**	**Linear**
2	4.13%	0.33%	5.3	0.42
3	5.47%	**0.21%**	7.38	**0.29**
4	5.20%	0.24%	6.13	0.30
5	5.69%	0.23%	6.70	0.26

Note: boldface indicates the best result.

Table 7. Clustering results of best KM-SVR (K = 3) for physical channel (Issue 1).

	Cluster 1	**Cluster 2**	**Cluster 3**	**Total**
Training data	61	54	29	144
Testing data	5	5	2	12
Total	66	59	31	156

4.3.2. Comparison of Prediction Models

The prediction results for physical channel (Issue 1 and 2) are summarized in Table 8. In a mere comparison of pure algorithms, which are served as the benchmarking models, simple SVR produced lower MAPE and RMSE than those of simple ELM. More importantly, the proposed KM-ELM and KM-SVR achieved better prediction accuracy than simple ELM, and simple SVR. KM-ELM was the best model (MAPE = 0.13%↓, RMSE = 0.14↓) when performing demand forecasting for Issue 1 and Issue 2 of the physical channel. Similarly, when constructing non-store demand forecasting models for Issue 3 and Issue 4, KM-ELM still performed the best, though by a low margin. Although RMSE of KM-ELM in Issue 4 is higher than that of KM-SVR by 0.01, it still has the lowest MAPE, as shown in Table 9.

Table 8. Physical channel results with 10 and 24 predictor variables.

		ELM	**SVR**	**KM-ELM**	**KM-SVR**
10-predictor forecasting model (Issue 1)	MAPE	0.44%	0.42%	**0.13%**↓	0.21%↓
	RMSE	0.62	0.59	**0.14**↓	0.29↓
24-predictor forecasting model (Issue 2)	MAPE	0.70%	0.68%	**0.50%**↓	0.57%↓
	RMSE	0.80	0.73	**0.64**↓	0.68↓

Note: boldface indicates the best results and ↓ denotes the decreases compared with the benchmark models.

Table 9. Non-store channel results with 10 and 24 predictor variables.

		ELM	SVR	KM-ELM	KM-SVR
10-predictor forecasting model (Issue 3)	MAPE	0.32%	0.42%	**0.14%**↓	0.26%↓
	RMSE	0.41	0.40	**0.15**↓	0.21↓
24-predictor forecasting model (Issue 4)	MAPE	0.31%	0.36%	**0.25%**↓	0.27%↓
	RMSE	0.40	0.40	0.23↓	**0.22**↓

Note: boldface indicates the best results and ↓ denotes the decreases compared with the benchmark models.

It implies that the proposed KM-ELM and KM-SVR methods are applicable and more effective for fast-fashion demand forecasting, regardless of retailing channels. Since fashion retailers periodically launch new products or remodel the existing merchandise on the markets to keep pace with the customer's craze and sensation, they accordingly have to orchestrate their marketing campaigns in a timely manner for implementation of annual sales plans over different seasons. As a result, the demand of fashion products exhibits similar data patterns or features at different time periods. Aligned with the inference above, KM-ELM and KM-SVR can obtain promising forecasting performances by means of clustering the data in several groups to reduce the likelihood that a model learns the patterns irrelevant to the testing data in the training phase.

However, the accuracy declined slightly when the number of predictor variables increased in a model. This could be attributable to the curse of dimensionality, which is an interesting topic for future studies.

5. Conclusions

The effective operation of a retailer relies on demand forecasting more than other strategic and planning decisions. An accurate demand forecast can directly affect the

company's profitability and competitive advantage in the market. This study uses the demand data of fashion retailer's physical and non-store channels as empirical data and constructs demand forecasting models through KM clustering combined with ELMs or SVR. The results show that KM-ELM and KM-SVR have higher prediction accuracy than the simple ELM and SVR prediction models. This verifies the position that, once data are sorted through clustering, the pre-processing of data homogenization can shorten the computational time needed to construct a model and help alleviate high complexity of demand data specific to the fashion industry. As a whole, clustering-based machine learning forecasting models can improve predictions under time pressure.

The fast fashion industry has uncertain product demand, extremely short product life cycles, and insufficient or incomplete historical data. However, constructing a forecasting model using machine learning requires high quality and sufficient data during the training process. Therefore, this study uses BIAS indicators, which are commonly used in time series measurement models, as predictor variables, and incorporates meteorological data as an attempt to compensate for the scarcity of data. It might be counterintuitive that the empirical results show the forecast results using 10 demand predictor variables to be more accurate than the forecast results using 24 demand predictor variables. We speculate that there are a few possible reasons for this. The empirical data used in this study are the rate of change in demand rather than the net demand and are the overall national demand rather than a local demand, which implies that they might not be affected by the climate of each specific region. Therefore, it is recommended that future researchers consider other external predictor variables such as holidays (school and business), commercially relevant events (Easter, Christmas), extraordinary events (e.g., a pandemic), and economic conditions when predicting demand for a more specific region or different sales channels. In addition, for the sake of obtaining more detailed managerial implications, interpretable machine learning algorithms can be the alterative methodology in future studies.

The results of this study show that the KM-ELM model is most suitable for demand forecasting in the fashion industry, and its superior forecasting ability makes it helpful for the operation and management of production and sales. Fast fashion companies can use the KM-ELM model even with limited data and under tight time constraints to obtain an accurate prediction result that is conducive to improving company performance management. This also enables fashion companies to achieve efficient and timely inventory management, which reduces inventory costs for retailers by lowering the probability of under- or overstocking.

Author Contributions: Conceptualization, I.-F.C.; methodology, I.-F.C. and C.-J.L.; investigation, I.-F.C. and C.-J.L.; writing: I.-F.C. and C.-J.L. Both authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by Ministry of Science and Technology, Taiwan, under grant number: 109-2622-E-030-001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This data can be found here: https://www.fastretailing.com.

Acknowledgments: We deeply appreciate Ting-Syuan Jhu's contributions to the pre-processing of data and technical implementation of models. The authors would like to thank the editor and anonymous reviewers for their valuable and constructive comments, which have led to a significant improvement in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Frings, G.S. *Fashion: From Concept to Consumer*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1982; p. 305.
2. Nenni, M.E.; Giustiniano, L.; Pirolo, L. Demand Forecasting in the Fashion Industry: A Review. *Int. J. Eng. Bus. Manag.* **2013**, *5*, 37. [CrossRef]

3. Yu, Y.; Choi, T.-M.; Hui, C.-L. An intelligent fast sales forecasting model for fashion products. *Expert Syst. Appl.* **2010**, *38*, 7373–7379. [CrossRef]
4. Hossain, M.M.; Abdulla, F. Forecasting the garlic production in Bangladesh by ARIMA Model. *Asian J. Crop Sci.* **2015**, *7*, 147.
5. Hsu, C.-C.; Chen, C.-Y. Applications of improved grey prediction model for power demand forecasting. *Energy Convers. Manag.* **2003**, *44*, 2241–2249. [CrossRef]
6. Choi, T.-M.; Hui, C.-L.; Liu, N.; Ng, S.-F.; Yu, Y. Fast fashion sales forecasting with limited data and time. *Decis. Support Syst.* **2014**, *59*, 84–92. [CrossRef]
7. Sun, Z.L.; Choi, T.M.; Au, K.F.; Yu, Y. Sales forecasting using extreme learning machine with applications in fashion re-tailing. *Decis. Support Syst.* **2008**, *46*, 411–419. [CrossRef]
8. Lu, C.J.; Shao, Y.E. Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learning machine. *Math. Probl. Eng.* **2012**, *2012*, 831201. [CrossRef]
9. Lu, C.-J. Sales forecasting of computer products based on variable selection scheme and support vector regression. *Neurocomputing* **2014**, *128*, 491–499. [CrossRef]
10. Yu, L.; Dai, W.; Tang, L. A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting. *Eng. Appl. Artif. Intell.* **2016**, *47*, 110–121. [CrossRef]
11. Chen, I.-F.; Lu, C.-J. Sales forecasting by combining clustering and machine-learning techniques for computer retailing. *Neural Comput. Appl.* **2016**, *28*, 2633–2647. [CrossRef]
12. Tan, J.Y.B.; Bong, D.B.L.; Rigit, A.R.H. Time series prediction using backpropagation network optimized by Hybrid K-means-Greedy Algorithm. *Eng. Lett.* **2012**, *20*, 203–210.
13. Benmouiza, K.; Cheknane, A. Forecasting hourly global solar radiation using hybrid K-means and nonlinear autoregressive neural network models. *Energy Convers. Manag.* **2013**, *75*, 561–569. [CrossRef]
14. Lu, C.J.; Lee, T.S.; Lian, C.M. Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks. *Decis. Support Syst.* **2012**, *54*, 584–596. [CrossRef]
15. Ramos, P.; Santos, N.; Rebelo, R. Performance of state space and ARIMA models for consumer retail sales forecasting. *Robot. Comput. Manuf.* **2015**, *34*, 151–163. [CrossRef]
16. Murat, M.; Malinowska, I.; Gos, M.; Krzyszczak, J. Forecasting daily meteorological time series using ARIMA and regression models. *Int. Agrophysics* **2018**, *32*, 253–264. [CrossRef]
17. Chen, S.; Chen, J. Forecasting container throughput at ports using genetic programming. *Expert Syst. Appl.* **2010**, *37*, 2054–2058. [CrossRef]
18. Frank, C.; Garg, A.; Sztandera, L.; Raheja, A. Forecasting women's apparel sales using mathematical modeling. *Int. J. Cloth. Sci. Technol.* **2003**, *15*, 107–125. [CrossRef]
19. Au, K.-F.; Choi, T.-M.; Yu, Y. Fashion retail forecasting by evolutionary neural networks. *Int. J. Prod. Econ.* **2008**, *114*, 615–630. [CrossRef]
20. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965; Volume 1, pp. 281–297.
21. Guftar, M.; Ali, S.H.; Raja, A.A.; Qamar, U. A novel framework for classification of syncope disease using K-means clustering algorithm. In Proceedings of the 2015 SAI Intelligent Systems Conference (IntelliSys), London, UK, 10–11 November 2015; pp. 127–132.
22. Xue, W.; Feijia, L.; Wenxia, X.; Kun, G.; Guodong, L. Based on K-Means Clustering and CNN Algorithm Research in Hail Cloud Determination. In Proceedings of the 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, Nanchang, China, 13–14 June 2015; pp. 232–235.
23. Huang, G.-B.; Chen, L.; Siew, C.-K. Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef]
24. Benoît, F.; van Heeswijk, M.; Miche, Y.; Verleysen, M.; Lendasse, A. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing* **2013**, *102*, 111–124. [CrossRef]
25. Huang, G.; Song, S.; Gupta, J.N.D.; Wu, C. Semi-Supervised and Unsupervised Extreme Learning Machines. *IEEE Trans. Cybern.* **2014**, *44*, 2405–2417. [CrossRef]
26. Kasun, L.L.C.; Zhou, H.; Huang, G.B.; Vong, C.M. Representational learning with extreme learning machine for big data. *IEEE Intell. Syst.* **2013**, *28*, 31–34.
27. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61*, 32–48. [CrossRef] [PubMed]
28. Xia, M.; Lu, W.; Yang, J.; Ma, Y.; Yao, W.; Zheng, Z. A hybrid method based on extreme learning machine and k-nearest neighbor for cloud classification of ground-based visible cloud image. *Neurocomputing* **2015**, *160*, 238–249. [CrossRef]
29. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
30. Shrivastava, N.A.; Panigrahi, B.K. A hybrid wavelet-ELM based short term price forecasting for electricity markets. *Int. J. Electr. Power Energy Syst.* **2014**, *55*, 41–50. [CrossRef]
31. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer, Inc.: New York, NY, USA, 1995.

32. Yu, X.; Qi, Z.; Zhao, Y. Support Vector Regression for Newspaper/Magazine Sales Forecasting. *Procedia Comput. Sci.* **2013**, *17*, 1055–1062. [CrossRef]
33. Du, X.F.; Leung, S.C.; Zhang, J.L.; Lai, K.K. Demand forecasting of perishable farm products using support vector machine. *Int. J. Syst. Sci.* **2013**, *44*, 556–567. [CrossRef]
34. Gunn, S.R. Support vector machines for classification and regression. *ISIS Tech. Rep.* **1998**, *14*, 5–16.
35. Hsu, C.W.; Chang, C.C.; Lin, C.J. *A Practical Guide to Support Vector Classification*; National Taiwan University: Taipei, Taiwan, 2003; pp. 1–16.
36. Cai, X.; Nan, X.Y.; Gao, B.P. Oxygen supply prediction model based on IWO-SVR in bio-oxidation pretreatment. *Eng. Lett.* **2015**, *23*, 173–179.
37. Kim, S.; Kim, H. A new metric of absolute percentage error for intermittent demand forecasts. *Int. J. Forecast.* **2016**, *32*, 669–679. [CrossRef]
38. Willmott, C.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]
39. Lewis, E.B. Control of body segment differentiation in drosophila by the bithorax gene complex. *Prog. Clin. Biol. Res.* **1982**, *1*, 269–288.
40. Lu, C.J.; Chen, I.F. Identifying important predictors for computer server sales using an effective hybrid forecasting technique. *Int. J. Inf. Manag. Sci.* **2017**, *28*, 213–232.
41. Lu, C.J.; Kao, L.J. A clustering-based sales forecasting scheme by using extreme learning machine and ensembling link-age methods with applications to computer server. *Eng. Appl. Artif. Intell.* **2016**, *55*, 231–238. [CrossRef]
42. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Cañar Benavides, José Alejandro** con C.C: #**0705036069** autor/a del trabajo de titulación: **Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning,** previo a la obtención del título de **Licenciado de Negocios Internacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **22** de **agosto** de **2026**

f. 

Cañar Benavides, José Alejandro

C.C: #**0705036069**

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Iglesias Muñoz, Valeria Abigail** con C.C: # **1722211677** autor/a del trabajo de titulación: **Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning,** previo a la obtención del título de **Licenciado de Negocios Internacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **22** de **agosto** de **2026**

f. _____

Iglesias Muñoz, Valeria Abigail

C.C: # **1722211677**

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TEMA Y SUBTEMA:	**Estimación de ventas mediante un algoritmo de K Means para la predicción basados en Machine Learning**		
AUTOR(ES)	**Cañar Benavides, José Alejandro e Iglesias Muñoz, Valeria Abigail.**		
REVISOR(ES)/TUTOR(ES)	**Ing. Carrera Buri, Félix Miguel, Mgs**		
INSTITUCIÓN:	**Universidad Católica de Santiago de Guayaquil**		
FACULTAD:	**Facultad de Economía y Empresa**		
CARRERA:	**Negocios Internacionales**		
TITULO OBTENIDO:	**Licenciado en Negocios Internacionales**		
FECHA DE PUBLICACIÓN:	**22 de agosto de 2026**	**No. DE PÁGINAS:**	**116 p.**
ÁREAS TEMÁTICAS:	Inteligencia de Negocios, Machine Learning, Producción acuícola,		
PALABRAS CLAVES/ KEYWORDS:	Estimación de ventas, K-Means, machine learning, regresión lineal múltiple, alimento balanceado, sector camaronero		
RESUMEN/ABSTRACT (150-250 palabras): Este trabajo investiga la estimación de ventas de alimento balanceado para camarón mediante la combinación del algoritmo K-Means y un modelo de regresión lineal múltiple, bajo un enfoque cuantitativo, aplicado, descriptivo y predictivo. El estudio se realizó con una base histórica de 7.216 registros y 38 variables correspondientes a transacciones comerciales realizadas entre enero de 2025 y mayo de 2026. El procesamiento se realizó en Python mediante Google Colab, incluyendo inspección, limpieza, manejo de valores faltantes y atípicos, conversión de tipos, eliminación de variables irrelevantes, codificación One-Hot y estandarización. Se utilizó K-Means para realizar el agrupamiento y se evaluó el número óptimo de clústeres mediante el método del codo y el coeficiente de silueta, determinándose que la mejor configuración fue con dos grupos. Posteriormente, se incluyó la etiqueta Clúster como variable explicativa dentro de una regresión lineal múltiple con el fin de estimar Total USD. El modelo obtuvo un R^2 de 0,9883, un error cuadrático medio de 710.081,00 y un error absoluto medio de 485,65. Los resultados mostraron que los valores observados y estimados coincidieron en un alto grado, aunque se encontraron desviaciones mayores en las transacciones negativas o cercanas a cero. Se concluye que la combinación de K-Means y regresión lineal múltiple permite reconocer patrones transaccionales y generar estimaciones útiles para apoyar decisiones relacionadas con inventarios, compras, despachos, metas comerciales y recursos financieros, considerando las condiciones metodológicas específicas del modelo aplicado. La investigación también comparó sus resultados con estudios anteriores sobre segmentación y predicción comercial, encontrando coincidencias en la utilidad analítica de estas técnicas aplicadas.			
ADJUNTO PDF:	☒ SI	☐ NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-984274426 +593-93 969 7891	E-mail: iglesias.valeria.m@gmail.com josealejandrocabe10@gmail.com	
CONTACTO CON LA INSTITUCIÓN (C00RDINADOR DEL PROCESO UTE)::	**Nombre: Freire Quintero Cesar Enrique**		
	Teléfono: +593- 990090702		
	E-mail: cesar.freire@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			