



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

TÍTULO:

**“Optimización del stock de los productos de Toalco mediante un algoritmo
K-means”**

AUTORES:

Castro Perrone Franccesca Valentina

Párraga Rosales Jostyn Josué

**Trabajo de Integración curricular previo a la obtención de
título de LICENCIADO EN NEGOCIOS
INTERNACIONALES**

TUTOR:

Ing. Carrera Buri, Félix Miguel, Mgs.

Guayaquil, Ecuador

22 agosto de 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 Castro Perrone, Franccesca Valentina y Párraga Rosales, Jostyn Josué como requerimiento para la obtención del título de Licenciados en Negocios Internacionales.

f. _____

Ing. Carrera Buri, Félix Miguel, Mgs

TUTOR

f. _____

Ing. Hurtado Cevallos, Gabriela Elizabeth, Mgs

DIRECTOR DE LA CARRERA

Guayaquil, a los 22 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, Castro Perrone, FrANCESCA Valentina
Párraga Rosales, Jostyn Josué

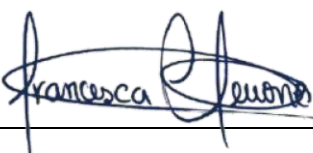
DECLARAMOS QUE:

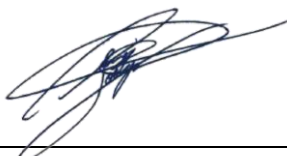
El Trabajo de Titulación, **Optimización del stock de los productos de Toalco mediante un algoritmo K-means**, previo a la obtención del título de Licenciados en Negocios Internacionales, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, 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 del mes de agosto del año 2026

LOS AUTORES:

f. 
Castro Perrone, Francesca Valentina

f. 
Párraga Rosales, Jostyn Josué



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

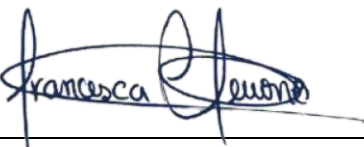
AUTORIZACIÓN

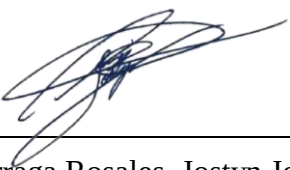
Nosotros, Castro Perrone, Franccesca Valentina
Párraga Rosales, Jostyn Josué

Autorizamos a la Universidad Católica de Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación, **Optimización del stock de los productos de Toalco mediante un algoritmo K-means**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

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

LOS AUTORES:

f. 
Castro Perrone, Franccesca Valentina

f. 
Párraga Rosales, Jostyn Josué

FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

REPORTE COMPILATIO



Certificado de análisis

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

Optimización del stock de los productos de Toalco mediante un algoritmo - Castro Perrone & Párraga Rosales

ID : 62f42f7178ed8497d83e95a3c67d4247f87b9b50



0%

Textos
sospechosos

Nombre del fichero : Optimización del stock de los productos de Toalco
mediante un algoritmo - Castro Perrone & Párraga Rosales.txt
Tamaño del archivo original : 9,17 MB
Número de palabras : 37.297

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

 Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

f. _____

Ing. Carrera Buri, Félix Miguel, Mgs
TUTOR

f. _____

Ing. Hurtado Cevallos, Gabriela Elizabeth, Mgs
DIRECTOR DE LA CARRERA

Agradecimiento

A mi fuente de amor infinito, por acompañarme durante este camino, por creer en mí incluso cuando dudaba, por celebrar cada logro y alentarme cada que lo necesito.

A ti, mi Eleonor, gracias por ser la luz de mi vida y mi amor más grande. Gracias por devolverle el sentido a mi vida.

A mis amigos, por su compañía incondicional en cada etapa de mi vida.

A las personas que conocí durante este proceso, a quienes se quedaron e hicieron que este camino sea más llevadero.

A las personas que fueron luz en mi vida en el momento preciso, gracias por ver más allá de los ojos.

A mi tutor, Félix Carrera, por su orientación y apoyo constante durante el desarrollo de este trabajo.

Franccesca Valentina Castro Perrone

Dedicatoria

A mi abuelo, por su amor que trascendió pero que nunca se fue, por ser mi guía y mi fortaleza en todo momento. Gracias por seguirme acompañando pese a la distancia que la vida nos impuso. Tu esencia vive en mí, guiando mis pasos y recordándome que el verdadero amor trasciende cualquier distancia.

Francesca Valentina Castro Perrone

Agradecimiento

Primeramente, a Dios por haberme guiado en todo momento. A mi familia eternamente ya que siempre confiaron en que podía lograrlo a pesar de las dificultades y adversidades que nos depara la vida.

A mi madre Diana Rosales por ser la guía y motor en mi vida, por siempre ser la primera en despertar y la última en dormirse, por siempre estar al pendiente de mí y de todas las desveladas de estudio en los primeros semestres. Estoy eternamente agradecido por tener a una madre tan valiente que me enseñó a no rendirme jamás; tu fortaleza y perseverancia son virtudes que admiro profundamente. Te amo, madre.

A mi abuelo, por su amor incondicional, gracias por creer en mí, por tus bendiciones antes de salir de casa y por acompañarme con tu cariño infinito en cada paso de este camino.

A mi hermana Diana Párraga por estar conmigo en todo este camino, gracias por ser una fuente de inspiración de constancia y esfuerzo. Asimismo, a mis hermanos y familia por su apoyo incondicional y por siempre a mi lado a pesar de la distancia.

A mis amigos de siempre, por ser mi espacio de tranquilidad y desconexión durante todos estos años. Gracias por entender mis ausencias cuando el estudio o el trabajo me lo exigían, y por estar cerca brindándome su apoyo incondicional.

A las maravillosas personas que la universidad me ha permitido conocer, a mis compañeros, docentes, asesores. Muchas gracias por ser parte de este camino siempre los llevare conmigo.

Para concluir, un agradecimiento profundo a Guissella Merchán Calderón por ser un pilar desde el día uno de carrera por ser mi espacio seguro, por escucharme con empatía y por ayudarme a encontrar claridad en medio de la incertidumbre. Sinceramente, sin su guía y su ayuda, alcanzar esta meta no habría sido posible.

Jostyn Josué Párraga Rosales

Dedicatoria

A ti, papá, que sé que estarías muy orgulloso de verme cerrar esta etapa de mi vida. Aunque tu partida en segundo semestre me frenó por un momento, también me dio la fuerza para avanzar y seguir adelante, a pesar de lo difícil que era volver a clases. Gracias por acompañarme todo este tiempo; sé que no te has ido, porque sigues vivo en mi corazón y en cada uno de mis recuerdos.

Gracias por enseñarme a ser paciente, a nunca dudar de mí, a saber escuchar, y por esa gran lección: que para todo hay una solución en la vida, excepto para la muerte.

Jostyn Josué Párraga Rosales



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

TRIBUNAL DE SUSTENTACIÓN

f. _____

**Ing. Hurtado Cevallos, Gabriela Elizabeth, Mgs.
DIRECTORA DE CARRERA**

f. _____

**PhD. Cesar Enrique Freire Quintero
COORDINADOR DEL ÁREA**

f. _____

OPONENTE



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

CALIFICACIÓN

Castro Perrone, Francческа Valentina

Parraga Rosales, Jostyn Josue

1 TABLA DE CONTENIDO

2	Introducción	2
3	Problemática	7
4	Justificación.....	11
5	Alcance.....	13
6	Objetivos.....	14
6.1	Objetivo general.....	14
7	Marco Teórico.....	15
7.1	Productos para acabados de construcción.....	15
7.2	Gestión de inventario	16
7.3	Optimización del stock	17
7.4	Inteligencia Artificial.....	18
7.4.1	Inteligencia Artificial aplicada al sector de materiales para construcción.....	19
7.5	Machine Learning.....	20
7.5.1	Aprendizaje no supervisado.....	23
7.6	Análisis de Componentes Principales.....	24
7.7	K-Means	27
7.7.1	Aplicación de K-Means en la Optimización de Stock.....	30
8	Marco conceptual.....	32
8.1	Inteligencia Artificial.....	32
8.1.1	Machine Learning.....	32
8.1.2	Modelo no supervisado.....	33
8.2	Análisis de Componentes Principales.....	33
8.3	K-Means	35
8.3.1	Clustering.....	36

9	Marco Legal	37
10	Metodología	40
10.1	Fuente de datos	40
10.2	Tratamiento de los datos	40
10.3	Preparación Inicial de Datos	41
10.3.1	Creación del Maestro de Productos (máster)	41
10.4	Variables de Demanda	41
10.4.1	Demanda Total	41
10.4.2	Demanda	43
10.4.3	Intermitencia	45
10.5	Análisis de Componentes Principales (ACP)	48
10.5.1	Estandarización Z (StandardScaler)	48
10.5.2	Eigenvectores y Eigenvalores (Análisis de Componentes Principales)	50
10.5.3	Covarianza	52
10.6	Ortogonalidad de los Componentes Principales	53
10.6.1	Reducción de Dimensionalidad (Varianza Explicada)	53
10.7	Criterio de Kaiser	54
10.8	Agrupamiento K-Means	55
10.8.1	Método del Codo (Elbow Method)	55
10.9	Función Objetivo K-Means (Inercia)	57
10.10	Métricas de Evaluación de Clustering	62
10.10.1	Perfilado de Clústeres	62
11	METODOLOGÍA	64
11.1	ESTADÍSTICO DESCRIPTIVO	64
11.2	Variables categóricas	65

11.3	Variable numérica.....	65
12	Análisis de resultados	67
12.1	Caracterización descriptiva de la base de datos.....	67
12.2	Variable Artículo (Producto)	67
12.3	Variable Línea	68
12.4	Variable Documento	70
12.5	Variable Persona (Compradores).....	71
12.6	Variable numérica Precio	73
12.6.1	Medidas de resumen	74
12.7	Construcción y ejecución del modelo.....	75
12.8	Paso 1.- Preparación del Entorno de Trabajo	76
12.8.1	Módulos de aprendizaje automático	77
12.9	Paso 2: Carga y limpieza de los datos	78
	Se agrega el archivo con los datos	78
12.9.1	Transferencia del archivo al entorno de trabajo de Google Colab	78
12.9.2	Lectura y normalización de encabezados	79
12.9.3	Depuración de datos duplicados	79
12.10	Paso 3: Construcción de las variables.....	80
12.10.1	Conversión de tipos y tabla maestra de productos.....	80
12.10.2	Matriz de demanda mensual por producto.....	81
12.10.3	Variables de demanda, variabilidad e intermitencia.....	82
12.10.4	Variables de saldo y rotación.....	82
12.10.5	Costo unitario mediante cascada de validez	83
12.10.6	Consolidación de la matriz de características	83
12.11	Paso 4: Transformación, Partición y Estandarización	84

12.11.1	Definición de variables y transformaciones	84
12.11.2	Partición en subconjuntos de entrenamiento y prueba	85
12.11.3	Tratamiento de valores faltantes.....	86
12.11.4	Estandarización de variables.....	87
12.12	Paso 5: Reducción de Dimensionalidad	87
12.12.1	Ajuste del Análisis de Componentes Principales	87
12.12.2	Matriz de cargas factoriales y proyección de los datos	88
12.12.3	Representación gráfica de la varianza explicada	88
12.12.4	Representación gráfica de las cargas factoriales	91
12.12.5	Celda de registro de artefactos.....	92
12.13	Paso 6: Segmentación mediante K-means.....	92
12.13.1	Evaluación iterativa del número de conglomerados.....	92
12.13.2	Ajuste del modelo definitivo	93
12.13.3	Representación gráfica de los criterios de selección	93
12.14	Paso 7: Evaluación y Perfilamiento.....	96
12.14.1	Métricas de validación interna.....	96
12.14.2	Consolidación de resultados y perfilamiento.....	97
12.14.3	Presentación del perfil de conglomerados	97
12.14.4	Visualización de los conglomerados	99
12.14.5	Celda final de registro de artefactos	100
12.15	Interpretación de los Resultados.....	101
12.15.1	Conglomerado 2: productos críticos de alto valor	101
12.15.2	Conglomerado 3: productos de rotación masiva	102
12.15.3	Conglomerado 0: productos de demanda intermitente y valor intermedio.....	102
12.15.4	Conglomerado 1: productos de baja rotación y bajo valor	102
12.16	Consideraciones generales.....	103
13	Discusión.....	104
14	Conclusiones.....	112
15	Referencias	114
16	Anexos.....	117

16.1	Anexo 1. Base de datos.....	117
16.2	Anexo 2 Script en Python	121

Índice de Ilustraciones

Ilustración 1 Subcampos de la inteligencia artificial. Fuente: Elaboración propia	19
Ilustración 2 Tipos de aprendizajes del Machine Learning. Fuente: Elaboración propia	22
Ilustración 3 Proceso de agrupamiento mediante el algoritmo K-Means, desde la selección del número de clústeres hasta la obtención del resultado final. Fuente: Elaboración propia.....	29
Ilustración 4 Diagrama de flujo del algoritmo K-means. Obtenido de "Prediction of leukemia by classification and clustering techniques	57
Ilustración 5 Ilustración del proceso iterativo de asignación y actualización de centroides en K-Means. Adaptado de “K-means” por C. Piech, Stanford University CS221 Handouts.	60
Ilustración 6 Frecuencia relativa del Top 10 de productos y «Otros».	68
Ilustración 7 Distribución de frecuencia por Línea.	69
Ilustración 8 Frecuencia relativa por tipo de Documento.	70
Ilustración 9 Frecuencia relativa del Top 10 de compradores y «Otros».....	72
Ilustración 10 Frecuencia relativa por intervalos de Precio.	73
Ilustración 11 Histograma y densidad (KDE) de la distribución del Precio.	74
Ilustración 12: Medidas descriptivas de las variables numéricas	74
Ilustración 13: Medidas descriptivas de las variables numéricas. Fuente: Elaboración propia.	75
Ilustración 14 Varianza explicada por cada componente principal	90
Ilustración 15 Varianza explicada acumulada de los componentes principales	90
Ilustración 16 Cargas factoriales de las variables originales en los dos componentes principales	91
Ilustración 17 Inercia del algoritmo K-means para valores de k entre 2 y 8.....	95
Ilustración 18 Coeficiente de silueta promedio para valores de k entre 2 y 8.....	95
Ilustración 19 Conglomerados de productos en el espacio de componentes principales.	100
Ilustración 20 Proyección PCA de la separación de clústeres obtenida en Anggraini, Amali y Anwar (2026)	105
Ilustración 21 Ejemplos de resultados de los productos clusterizados.	106

Ilustración 22 Algoritmo MiniBatch K-means..... 109

Ilustración 23 Coeficientes de silueta..... 110

Índice de Tablas

Tabla 1: Matriz de conceptos y aplicaciones en Inteligencia Artificial y métodos clustering. Fuente:
Elaboración propia. 31

Tabla 3: Tipos de variables. Fuente: Elaboración propia. 64

Tabla 4: Frecuencias de los 10 productos más vendidos. Fuente: Elaboración propia. 67

Tabla 5: Frecuencias de las líneas de productos. Fuentes: Elaboración propia. 68

Tabla 6: Frecuencias por tipos de documentos. Fuentes: Elaboración propia. 70

Tabla 7: Frecuencias de compradores. Fuente: Elaboración propia. 71

Tabla 8: Frecuencias por intervalo de precios. Fuentes: Elaboración propia. 73

Tabla 9: Perfil promedio de los conglomerados obtenidos. Fuentes: Elaboración propia. 101

Tabla 10: Análisis de silueta de los clústeres obtenidos. Fuentes: Anggraini, Amali y Anwar (2026). 107

Resumen

La gestión correcta de inventarios es parte fundamental en la competitividad de las empresas en la actualidad, donde la innovación y la automatización de procesos mediante el análisis de datos se ha convertido en una herramienta efectiva para mejorar la clasificación de los productos, identificando las variables que influyen directamente en el comportamiento de este. Pese a ello, el entendimiento y la utilización de conceptos de aprendizaje automático, como lo son el Análisis de Componentes Principales y el algoritmo K-Means para la segmentación de productos, siguen siendo un desafío.

La clasificación de los productos y la identificación de las variables que afectan su rotación dentro del inventario son de principal importancia, ya que gracias a esto las empresas pueden adoptar estrategias adecuadas de aprovisionamiento y reducir los riesgos de sobrestock y desabastecimiento. La presente investigación se enfoca en el desarrollo de una metodología basada en el Análisis de Componentes Principales y el algoritmo K-Means para la segmentación de productos de inventario, en la cual se espera lograr una agrupación representativa de los productos según su comportamiento de demanda, costo y rotación.

La aplicación del Análisis de Componentes Principales y del algoritmo K-Means para esta segmentación se justifica debido a que la gestión de inventarios constituye uno de los procesos operativos más determinantes en la rentabilidad y la eficiencia de las empresas, al representar una parte significativa de su capital de trabajo.

Palabras clave: Inventario, Segmentación de productos, Análisis de Componentes Principales, K-Means, Aprendizaje no supervisado, Optimización de stock, Gestión de inventarios

Abstract

Efficient inventory management is fundamental to corporate competitiveness today, as innovation and process automation—driven by data analysis—have become effective tools for improving product classification by identifying the variables that directly influence product behavior. Nevertheless, understanding and applying machine learning concepts—such as Principal Component Analysis and the K-Means algorithm for product segmentation—remain a challenge.

The classification of products and the identification of the variables that affect their turnover and cost within the inventory are of primary importance, as they allow companies to adopt new procurement strategies and reduce the risks of overstock and stockouts. The present research focuses on developing a methodology based on Principal Component Analysis and the K-Means algorithm for the segmentation of inventory products, with the aim of achieving a representative grouping of products according to their demand, cost, and turnover behavior.

The application of Principal Component Analysis and the K-Means algorithm for this segmentation is justified given that inventory management constitutes one of the most decisive operational processes for the profitability and efficiency of companies, as it represents a significant share of their working capital.

Keywords: Inventory, Product segmentation, Principal Component Analysis, K-Means, Unsupervised learning, Stock optimization, Inventory management.

Résumé

Une gestion efficace des stocks est un élément fondamental de la compétitivité des entreprises d'aujourd'hui ; l'innovation et l'automatisation des processus fondées sur l'analyse des données sont devenues des outils performants pour améliorer la classification des produits en identifiant les variables qui influencent directement leur comportement. Toutefois, la compréhension et l'utilisation de concepts d'apprentissage automatique — tels que l'analyse en composantes principales et l'algorithme K-Means pour la segmentation des produits — restent un défi.

La classification des produits et l'identification des variables qui affectent leur rotation et leur coût au sein de l'inventaire sont primordiales, car elles permettent aux entreprises d'adopter de nouvelles stratégies d'approvisionnement et de réduire les risques de surstockage et de rupture de stock. La présente recherche se concentre sur le développement d'une méthodologie fondée sur l'Analyse en Composantes Principales et l'algorithme K-Means pour la segmentation des produits d'inventaire, dans le but d'obtenir un regroupement représentatif des produits selon leur comportement de demande, de coût et de rotation.

L'application de l'Analyse en Composantes Principales et de l'algorithme K-Means pour cette segmentation se justifie par le fait que la gestion des stocks constitue l'un des processus opérationnels les plus déterminants pour la rentabilité et l'efficacité des entreprises, représentant une part importante de leur fonds de roulement.

Mots-clés: Inventaire, Segmentation des produits, Analyse en Composantes Principales, K-Means, Apprentissage non supervise, Optimisation des stocks, Gestion des stock.

2 INTRODUCCIÓN

Actualmente el sector de la construcción constituye una de las actividades económicas con más auge a nivel mundial y nacional, dado su impacto en el desarrollo de infraestructura, urbano y dinamización comercial. Dentro de este sector las empresas dedicadas a la fabricación y comercialización de materiales y acabados para la construcción desempeñan un papel fundamental en el abastecimiento de productos destinados a proyectos residenciales comerciales e industriales.

La industria de la construcción representa uno de los sectores económicos más significativos a nivel mundial. Esto se debe, en parte, al crecimiento del sector urbano, el progreso de la infraestructura y generación de empleo. En los últimos años, este sector ha experimentado una modificación fomentada principalmente por la urbanización apresurada y la emergente necesidad de zonas residenciales y comerciales más viables.

Según el Banco Mundial, "Hoy en día, más de la mitad de la población mundial vive en ciudades. Se prevé que esta tendencia continúe y que la población urbana se duplique con creces para 2050, donde casi 7 de cada 10 personas vivirán en ciudades" (BancoMundial, s.f.).

Siendo este uno de los principales motivos por el cual el sector de la construcción está en su apogeo, el mismo mantiene proyecciones de crecimiento constante hacia los próximos años, esto la consolida como una actividad fundamental para economías desarrolladas y emergentes.

Según el informe "Global Powers of Construction 2024", elaborado por Deloitte, las 100 empresas constructoras más grandes del mundo generaron ingresos aproximados de 1.978 billones de dólares en 2024, esto evidencia la relevancia de esta industria en la economía mundial. Asimismo, menciona que se espera que el mercado global de la construcción siga en crecimiento para el año 2030.

Bajo este escenario, el segmento de productos de acabados de construcción presenta un importante alcance, esto es, en parte gracias a que se buscan materiales que no solo cumplan funciones estructurales, sino que puedan aportar tanto funcionalidad, como calidad y contribuir al valor estético. Buscando así diferenciarse en los existentes proyectos arquitectónicos.

En este segmento, los productos para acabados de construcción como cementos especializados, porcelanatos, revestimientos, pinturas, etc. Se han convertido en elementos esenciales para mejorar tanto el acabado como la apariencia en las construcciones modernas.

Esto en conjunto con las nuevas tendencias que cambian constantemente las preferencias del mercado. Podemos notar que, en la actualidad, se busca materiales sostenibles, amigables con el medio ambiente y que puedan otorgar un acabado más "estético"

En América Latina, el escenario de la industria de la construcción tiene también un impacto significativo en el desarrollo económico y social, esto se debe al efecto en la generación de empleo, y, que se considera también un reactivador en la economía porque acarrea numerosos sectores hacia atrás y hacia adelante.

Durante los últimos años, gracias a los factores ya mencionados como la demanda de viviendas, edificaciones comerciales y los proyectos de edificaciones de infraestructura pública que en parte son fomentados por el crecimiento poblacional, la migración, expansión de ciudades y también las nuevas tendencias emergentes. Estos factores han contribuido al fortalecimiento del mercado de materiales y productos para acabados de construcción en diversos países latinoamericanos.

Según la Comisión Económica para América Latina y el Caribe, América Latina continúa enfrentando fuertes retos con la deficiencia habitacional y la necesidad de innovar parte de su infraestructura urbana. Esto ha tenido un impacto positivo para el aumento de la demanda de

materiales de acabados de construcción.

El sector de la construcción en Ecuador vive una etapa de recuperación sostenida, declarada por el gobierno como un motor clave para la economía. De acuerdo con el Índice de Expectativas de la Economía (IEE) del mes de enero de 2026, el sector de la construcción obtuvo 61,6 puntos, lo que representa un aumento de 16,1 puntos en relación con enero de 2025 (Ministerio de Economía y Finanzas, 2026).

Según el Banco Central del Ecuador, la economía del país mostró mejoría durante el año 2025. Esto ha sido gracias a que las exportaciones, la inversión y el consumo de los hogares incrementó. Este incremento económico ha permitido que se reactiven múltiples sectores, entre ellos la construcción.

En Ecuador, el sector de la construcción y los acabados desempeña un papel fundamental en la economía del país, debido a que aporta aproximadamente el 7% al Producto Interno Bruto (PIB). Esta relevancia se basa en la relación directa que mantiene con actividades como la arquitectura, ingeniería civil y diseño de interiores. Sin contar, que el sector de los acabados de construcción ha tomado más protagonismo debido al aumento de proyectos basados en la remodelación y renovación de edificaciones. Aunque este sector tuvo una caída del 12% aproximadamente durante el año 2020, años siguientes ha mostrado una gran recuperación aportando directamente al crecimiento económico generando empleo y atracción de nuevas inversiones.

Menciona Roberto Luque que se está potenciando los rendimientos de construcción de viviendas que se financian a través de programas estatales, Asimismo menciona que el producto "Miti-Miti" se originó con un índice de 250 unidades y buscar alcanzar un objetivo de 600 unidades por mes, el programa ha registrado hasta el momento 8759 viviendas suministradas a través de créditos desde el 4,99% a 30 años (Luque, 2026).

Bajo este parámetro se evaluó con datos oficiales de las más relevantes en el sector de la construcción, acero, cementos, ferreterías, constructoras, tuberías, pisos, acabados, entre otras donde han visto una expansión del sector de un 7,4%.

El sector ha superado la crisis energética e inseguridad que limitaron su crecimiento en años anteriores y actualmente se enfoca en fortalecer dicho proceso mediante acciones dirigidas a mejorar la eficiencia con una facturación mensual que actualmente redondea USD 400 millones, la construcción corrobora su rol como el "termómetro" de la economía ecuatoriana: cuando este sector crece aceleradamente el país progresa hacia un escenario de mayor prosperidad y fortalecimiento del mercado laboral.

Cabe mencionar que el sector de construcción y acabados de construcción representa el 7% del PIB del Ecuador, según datos del Banco Central, siendo así una de las 5 industrias con más auge en este rubro, por lo que se transforma en un factor con un gran impacto en la situación económica e industrial en Ecuador (Banco Central del Ecuador, 2025).

Dentro de este marco macroeconómico, la gestión eficiente de los recursos operativos se convierte en una prioridad estratégica para las empresas del sector. De acuerdo con (Meindl, 2008) la cadena de suministro y la administración del inventario constituyen factores determinantes en la competitividad de las organizaciones comerciales, especialmente en sectores donde la variedad de productos es amplia y la demanda presenta comportamientos variables a lo largo del tiempo.

En este contexto, muchas de las empresas dedicada a la fabricación y comercialización de productos de acabados de construcción, enfrentan el desafío de gestionar eficientemente un inventario diversificado, compuesto por porcelanatos, pegamentos cerámicos, empastes y polvos minerales, entre otros. La dificultad de mantener niveles óptimos de stock en un mercado tan dinámico como el de la construcción ecuatoriana hace necesaria la adopción de herramientas analíticas avanzadas que superen

los métodos tradicionales de gestión empírica (Ballou, 2004)

Es aquí, donde conceptos basados en Inteligencia Artificial como el aprendizaje automático o Machine Learning han emergido como una disciplina de alto potencial para transformar los procesos de gestión de inventarios. Sus algoritmos permiten analizar grandes volúmenes de datos históricos, identificar patrones de comportamiento en la demanda y segmentar productos según características operativas comunes, facilitando así una toma de decisiones más objetiva, ágil y sustentada en evidencia (Mitchell, 1997). En particular, el algoritmo K-means se ha posicionado como una de las herramientas más utilizadas para la segmentación de inventarios, dada su eficiencia computacional y su facilidad de interpretación práctica (M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela, 2012)

3 PROBLEMÁTICA

Las empresas dedicadas a la comercialización y distribución de productos de acabados de construcción hacen frente a constantes desafíos relacionados con la gestión de la administración eficiente de inventarios, esto se debe, en parte a la amplia gama de productos existentes en este nicho, variabilidad en la demanda, tiempos prolongados de abastecimiento y a la mala planificación de inventarios. Este tipo de situaciones puede provocar problemas de desabastecimiento o sobre stock, lo cual tiene un impacto negativo tanto en el área operativa, financiera y comercial de una organización

Sumado a ello muchas empresas en Ecuador afronta retos significativos relacionados con la gestión eficiente de stock. La carencia de controles óptimos conduce frecuentemente a mermas sustanciales de mercadería, lo cual tiene repercusiones desfavorables directo en los clientes, los ingresos y en la rentabilidad general. Sin un seguimiento optimo, las entidades corren el riesgo de provocar insatisfacción en los clientes.

Por lo general, el inventario puede llegar a ser el mayor activo en el balance de una empresa, los costos generados por los inventarios pueden llegar a representar uno de los más altos rubros de estas (Garrido , Cejas, 2017)

Una inoportuna gestión de inventario genera costos de almacenamiento, pérdidas monetarias por mantener productos de baja rotación, así también puede influir en los retrasos en entregas a clientes. Por otro lado, la carencia de herramientas que permitan mantener un control de inventario obstaculiza el reabastecimiento, reposición y distribución de productos.

(Sharon, 2022) menciona que: Una correcta gestión del inventario es esencial para poder sentar las bases de un negocio rentable, pero sobre todo eficiente, ya que esto permite evitar pérdidas provocadas por mantener almacenados productos de baja rotación, o, por el contrario, porque disminuye la posibilidad de desperdicios a lo largo del proceso de producción.

Las constantes fallas en el control de inventario tanto a corto como mediano plazo son producto de la desorganización y ausencia de mecanismos que permitan tener un mejor control sobre los mismos, tanto así que llegan a generar ineficiencia al momento de utilizar los recursos.

(Angulo-Rivera, 2022), mencionan que es crucial recurrir a la planificación y la estructuración de los procesos para en control interno para poder lograr el cumplimiento de las metas y sobre todo la optimización de los recursos.

Si bien es cierto, la gestión y manejo de inventarios puede volverse una actividad compleja ya que el comportamiento de la demanda es imprescindible. Si se maneja una buena planificación y proyección, puede dar resultados positivos.

En un sector altamente competitivo como lo es el de los acabados de construcción, la optimización del stock se convierte en un factor crucial para mejorar la operatividad y poder garantizar la disponibilidad de productos mediante una entrega inmediata consiguiendo así la satisfacción del cliente. Pese a esto, actualmente no todas las empresas del sector han logrado implementar modelos eficientes que permitan gestionar el stock a niveles adecuados para responder activamente a los requerimientos del mercado

Es por ello, que surge la necesidad de desarrollar un modelo que te permita optimizar el stock para para mejorar el control de este, reducir costos operativos y cumplir los plazos de entrega.

En Ecuador, existen empresas del sector que siguen realizando el abastecimiento y control de inventarios de forma empírica, respaldándose en proyecciones o estimaciones manuales o en experiencia previa, sin hacer uso de herramientas tecnológicas o herramientas que permitan analizar datos e identificar patrones de comportamiento de la demanda. Esto trae como resultado mantener almacenados productos de baja rotación durante mucho tiempo.

Ante este problema, nace la necesidad de aplicar herramientas de análisis inteligentes para

mejorar la gestión y manejo de inventario identificando patrones y segmentando productos. Bajo esta circunstancia, algoritmos basados en IA del Machine Learning como K-means representan una opción eficiente que permitirá agrupar productos en base a características similares.

(M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela, 2012) mencionan que este algoritmo debe ser bien empleado ya que, si no se entrena correctamente, los resultados pueden ser desfavorables ya que uno de los principales problemas de este puede llegar a ser la incorrecta selección de número de clústeres y esto trae consigo agrupaciones erróneas y una interpretación de datos incorrecta.

Asimismo, existen empresas que aplican este algoritmo sin realizar un correcto proceso de limpieza y tratamiento de datos, esto puede convertir a los datos en poco confiables, causando desviación o sobreajustes innecesarios afectando a la calidad de segmentación.

No solo el mal tratamiento de datos es un problema, también puede darse que la interpretación de resultados no se la realice de forma correcta, muchas veces no se cuenta con criterios técnicos o metodológicos adecuados para saber interpretar los resultados.

Ante esta situación se manifiesta en la inexistencia de un sistema de clasificación técnica del inventario que permita diferenciar los productos según su nivel de rotación, comportamiento de ventas y criticidad para la operación comercial. La empresa mantiene una cartera diversa de productos (pegamento cerámico, porcelana, porcelanato, empaste, blanqueadores y polvos minerales), cada uno con funcionamiento de demanda distintas, lo que complejiza aún más la toma de decisiones de reabastecimiento (Angulo-Rivera, 2022).

Esta carencia de criterios técnicos de segmentación provoca que las decisiones de compra y almacenamiento se enfoquen en la intuición del personal o en registros históricos no sistematizados, aumentando la probabilidad de contener productos de baja rotación inmovilizados durante largos periodos, mientras que artículos de alta demanda pueden agotarse en momentos críticos. Según (Jain,

2010), la calidad de los resultados de cualquier algoritmo de agrupamiento depende en primera instancia de la calidad de los datos de entrada, por lo que el tratamiento riguroso de la información histórica de inventario es un requisito previo e indispensable para garantizar la validez del modelo.

4 JUSTIFICACIÓN

Una adecuada optimización de inventarios mejora notablemente la percepción de las mercancías, obteniendo así una mejor visión en tiempo real de la disponibilidad de las existencias que estén disponibles en los lugares donde los clientes los buscan, ya sea en cualquier sucursal de su preferencia, almacén u online. Esto habilita anticipar las adversidades evitando rupturas de stock que pueden convertirse en pérdidas de ventas y afectar considerablemente la satisfacción del cliente. Además, centralizar la data facilita un mejor manejo de pedidos más rápidos y eficientes, respaldando que los productos estén disponibles para ser enviados.

En base a esta situación una visibilidad mejorada ayuda a prevenir el sobre stock, reduciendo los gastos de almacenamiento de productos innecesarios. Por lo consiguiente el análisis de datos y tendencias del mercado optimiza las órdenes de compra a proveedores, disminuyendo costes de abastecimiento. De igual modo, al minimizar errores de inventarios, como errores de ingreso o mermas de mercancía las empresas logran ahorros considerablemente significativos.

No obstante, a pesar de la evolución tecnológica muchos negocios persisten en el uso de sistemas de inventarios rotatorios apoyados en conteos físicos en periodos regulares, a pesar de, estas técnicas permiten determinar el stock final y calcula el costo de ventas, suelen mostrar limitaciones en agilidad y dinamismo para responder a las dinámicas cambiantes del mercado actual, distinguido por la carencia de respuestas rápidas y eficientes.

Frente a esta situación, es clave optimizar de manera minuciosa las políticas de inventario, ajustándolas conforme a los estándares necesarios. Tomar en cuenta los parámetros correctos como el periodo de revisión, el tiempo de entrega y el nivel de servicio objetivo para cada producto de manera individual

El aprendizaje automático o machine learning, emplea técnicas y algoritmos avanzados para

analizar grandes volúmenes de datos, identificar patrones que no siempre son evidentes mediante métodos tradicionales. Sus algoritmos permiten detectar tendencias y correlaciones entre los productos almacenados, segmentándolos según sus particularidades comunes como niveles de rotación, cantidad de venta o su tendencia dentro del inventario. Esto viabiliza la segmentación del stock, asimismo contribuye a una toma de decisiones más eficaz en relación con el abastecimiento, almacenamiento, estrategias de reasignación y control de existencias.

Bajo esta perspectiva los algoritmos de machine learning pueden contribuir considerablemente a las empresas a definir los niveles óptimos de inventario para diferentes productos. Esto se alcanza considerando variables como el tiempo de entrega, la estacionalidad y las limitaciones de costos.

Considerando lo anterior, se torna vital progresar hacia modelos analíticos y sustentados en datos, que faciliten la correcta optimización de los productos en función de sus peculiaridades y pautas de rotación en el inventario. Bajo esta perspectiva, el empleo de metodologías de aprendizaje automático, como el algoritmo K-means, denota una opción viable para segmentar los productos de forma óptima.

Indican que K-means es una de las herramientas más ágiles y eficientes para evitar el overstocking (exceso de stock) y los costos financieros por desuso, empleando la integración rápida del algoritmo para agrupar miles de datos operativos en tiempo real.

El presente trabajo de investigación se realizó con el fin de contribuir a la optimización de los niveles de inventario mediante la aplicación de [el algoritmo K-means].

5 ALCANCE

La aplicación de un algoritmo K-means para la optimización de stock puede ser relevante para varios entes, entre ellos:

Empresas que manejen inventario dentro de sus operaciones con el objetivo de dar a conocer los conceptos de la inteligencia de negocios y el algoritmo K-means permitiéndoles implementar procesos automatizados en cuyas actividades aún posean sistemas realizados manualmente, generando inconvenientes en múltiples áreas de la organización.

Profesionales dedicados al manejo de inventario, este tipo de herramientas pueden beneficiar directamente a quienes desarrollan sus actividades manejando inventario. Ya que aplicar este tipo de algoritmos de Machine Learning puede ayudar a optimizar tiempo y agilizar sus funciones.

Profesionales dedicados al mundo del análisis de datos mediante algoritmos de Machine Learning.

La tecnología avanza cada día, y con ello también las habilidades requeridas para competir en el entorno laboral, dentro de poco, las personas que dominen el análisis de datos en conjunto de herramientas como el Machine Learning van a ser quienes lideren y compitan directamente en el entorno labora

6 OBJETIVOS

6.1 Objetivo general.

Desarrollar un modelo de segmentación de productos mediante Análisis de Componentes Principales y el algoritmo K-means, que sirva como herramienta de apoyo para la optimización del stock de productos de Toalco.

Objetivos Específicos

1. Analizar los conceptos y teorías del uso del algoritmo K-means en la optimización de stock mediante la aplicación de ACP para la reducción de dimensiones en la revisión de literatura.
2. Diseñar una metodología basada en ACP y K-Means que permita reducir la dimensionalidad de los indicadores de inventario y agrupar los productos según patrones de comportamiento similares.
3. Evaluar la eficacia del algoritmo ACP y K-means en la optimización del stock mediante la comparación de resultados de estudios previos que utilizaron metodologías similares.

7 MARCO TEÓRICO

7.1 Productos para acabados de construcción

En el sector de la construcción, los productos para acabados representan una parte esencial dentro de esta industria ya que estos materiales son utilizados en la parte final de la obra ya sea para aportar acabados estéticos, funcionalidad, protección y permanencia a las edificaciones. Dentro de los principales productos de acabados podemos encontrar pinturas, revestimientos, porcelana, porcelanato, adhesivos, cemento blanco, entre otros elementos que aportan a un mejor acabado y calidad de los proyectos inmobiliarios.

En países en desarrollo la construcción es el sector económico más importante después de la agricultura. Esto se debe también a que la industria de la construcción se reconoce como uno de los sectores económicos con mayor relevancia a nivel mundial gracias a su capacidad en generar empleo, atraer inversión y aportar al desarrollo económico en los países. Este crecimiento ha permitido que el mercado para acabados, especialmente los relacionados con sustentabilidad hayan tenido crecimiento (Edmonds, 1979) .

La venta y distribución de productos para acabados de construcción también ha tenido un impacto por la universalización de mercados. Hoy en día, varios países tienen manejan su inventario con productos importados para abastecer su demanda, generando así oportunidades para empresas comercializadoras.

7.2 Gestión de inventario

La gestión de inventarios es el conjunto de etapas de supervisión y regularización de variabilidad en la cadena de suministros, con el propósito de asegurar un nivel de inventario activo del producto correcto, en el lugar y momento adecuado, al menor costo posible (Ballou, 2004). Desde un enfoque financiero, el inventario constituye uno de los activos circulantes más destacados en empresas mercantiles y de distribución, razón por la cual su optimización influye directamente en la ganancia y la liquidez de la empresa.

Según (Silver, 2016) se distinguen tres clases esenciales de stock según su rol dentro de la cadena de abastecimiento. El stock de ciclo abarca la demanda prevista entre dos órdenes de reabastecimiento seguidas y es función directa de la habitualidad de pedido. El stock de seguridad incorpora las variaciones inesperadas en la demanda y en los periodos de entrega de los proveedores, funcionando como un escudo frente a la incertidumbre. El stock especulativo se conforma en anticipación a crecimientos de precio o a limitaciones futuras de oferta.

El control de stock requiere de un monitoreo constante de inventario, la notificación a tiempo de sobre o sub-stock, y el origen de alertas que indiquen la reposición de productos. Los sistemas de datos corporativos (ERP) se consideran como los fundamentos tecnológicos repetitivos para este control, ya que se encuentran los registros de las entradas y salidas de un producto, el stock existente, los pedidos en curso y los registros históricos. Sin embargo, como señalan (Hazen, 2016) la mera asequibilidad del dato transaccional no asegura la generación de inteligencia operativa; para obtener valor analítico de los archivos históricos de un ERP se necesitan métodos de tratamiento de datos que trasciendan los reportes descriptivos estándar.

7.3 Optimización del stock

La industria mundial de materiales de acabados ejerce un papel determinante en el desarrollo urbano y la dinámica económica. (Bhandari, 2026) menciona que este mercado alcanzó los USD 1,320 mil millones en 2023 y se espera que ascenderá a USD 1,867 mil millones en 2032, con un crecimiento sostenido de 3.9 % anual.

A pesar de ello, el mismo crecimiento ha evidenciado fallas estructurales en la gestión logística del sector, especialmente en el control de inventarios, donde la escasa sistematización puede causar altos costos de almacenamiento, sobre stock, desabastecimiento y pérdida de competitividad.

La correcta gestión del inventario es uno de los determinantes más críticos de la eficiencia operativa y financiera en empresas de distribución. Un exceso de stock inmoviliza capital de trabajo, incrementa los costos de almacenaje y eleva el riesgo de vencimiento o depreciación de productos; un déficit de stock, por el contrario, genera pérdida de ventas, deterioro de la relación con clientes y ruptura de la cadena de abastecimiento (Edward A. Silver, 2016).

La optimización del stock es un procedimiento crucial dentro de la gestión de inventarios con la finalidad de sostener la demanda sin sobre abastecerse de productos de baja rotación que conllevan altos costos de almacenamiento. Uno de los principales objetivos es asegurar la disponibilidad inmediata de productos en el momento en que sean requeridos, esto ayuda a minimizar el capital inmovilizado y reduce los riesgos que puedan asociarse al exceso o escasez de inventario.

La gestión de inventarios ha tenido una transformación convirtiéndose de un proceso operacional a un elemento determinante, que proviene de la necesidad de mantener controles de calidad, trazabilidad, reducción de costos, optimización de recursos y una mejor rotación de inventario.

7.4 Inteligencia Artificial

La Inteligencia Artificial (IA) constituye un campo amplio de estudio dentro de las ciencias de la computación, orientado al desarrollo de sistemas capaces de ejecutar tareas que tradicionalmente han requerido inteligencia humana.

La Inteligencia Artificial ha emergido como un tema engorroso y exhaustivo que puede llegar a considerarlo como un tema complejo y extenso en muchos casos para analizar y estudiar. Esta área de conocimiento es el resultado de la constante exploración de los humanos con herramientas que logren imitar o realizar procesos sin necesidad de intervención humana. También se la conoce como un campo de la informática enfocada a la evolución de sistemas aptos de ejecutar tareas que, desempeñadas por seres humanos, demandarían capacidades cognitivas como el razonamiento, el aprendizaje y la organización. (Stuart Russell, 2008)

La Inteligencia Artificial representa una de las tecnologías con mayor auge y efecto dentro de las empresas durante las recientes décadas. Su evolución ha admitido que los sistemas informáticos sean aptos de realizar tareas que usualmente necesitaban de la presencia humana, tales como la detección de patrones, la clasificación de información, la predicción de comportamientos. El desarrollo de la IA ha incentivado significativas transformaciones en diversos sectores económicos, autorizando mejorar la productividad operativa, optimizar recursos y reforzar los procesos de planificación. Gracias a la aptitud de analizar grandes volúmenes de información, las empresas logran identificar tendencias, reducir incertidumbre y mejorar la calidad de sus decisiones estratégicas.

Durante el último periodo, la aplicación de herramientas apoyadas en inteligencia artificial ha obtenido gran relevancia dentro de las cadenas de suministro y los sistemas logísticos. Por lo tanto, las empresas recurren a estas tecnologías para gestionar inventarios, optimizar rutas de distribución, predecir comportamientos de la demanda y optimizar el uso de sus recursos operativos (Giovanna

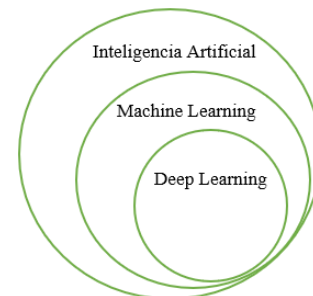
Culot, 2024).

En el seno de la industria de materiales para la construcción, la evolución digital ha motivado la aplicación continua de tecnologías inteligentes orientadas al control de operaciones y a la administración eficaz de recursos. El incremento de complejidad de las cadenas de abastecimiento, sumada a la necesidad de responder oportunamente a las variaciones del mercado, ha generado un escenario donde la inteligencia artificial se convierte en una herramienta de apoyo para la gestión empresarial.

En el escenario de la gestión empresarial, la IA ha avanzado de ser un campo de investigación académica a transformarse en un impulsor estratégico con aplicaciones probadas en la cadena de suministro, el control de calidad, la planificación de la producción y la gestión de inventarios.

Ilustración 1 Subcampos de la inteligencia artificial. Fuente: Elaboración propia

La implementación de la IA en la gestión de inventarios ha sido registrada en creciente precisión en los estudios de los últimos años. (Real Carbonneau, 2008) contrastaron el desempeño de redes neuronales, máquinas de soporte vectorial y modelos estadísticos clásicos en la predicción de la demanda en cadenas de suministro. Por su parte, (Aris A. Syntetos, 2016) analizó las bibliografías existentes sobre pronóstico de demanda para gestión de inventarios y destacaron que la implementación de técnicas de ML constituye el límite operativo del sector.



7.4.1 Inteligencia Artificial aplicada al sector de materiales para construcción

La industria de materiales para construcción se identifica por gestionar una amplia variedad de productos con diversos niveles de demanda, rotación y comportamiento comercial. Esta

diversidad produce relevantes inconvenientes relacionados con la gestión logística, el control de inventarios y la planificación de abastecimiento.

Durante el último periodo, las empresas del sector han empezado a integrar tecnologías basadas en inteligencia artificial con la finalidad de fortalecer la toma de decisiones y mejorar el desempeño operativo. Estas herramientas permiten analizar grandes cantidades de información proveniente de ventas, inventarios, proveedores y operaciones logísticas, facilitando la identificación de patrones difíciles de detectar mediante métodos tradicionales (Ellis, 2026)

La incorporación de inteligencia artificial en la gestión de materiales ha contribuido al desarrollo de sistemas más eficientes para la clasificación de productos, la optimización de inventarios y la segmentación de artículos según características específicas de comportamiento. Estas aplicaciones permiten que las organizaciones asignen de manera más eficiente sus recursos, mejoren la disponibilidad de productos y reduzcan costos asociados al almacenamiento excesivo (Cuervo, 2021).

De igual manera, la creciente digitalización del sector ha generado una mayor disponibilidad de datos operativos, lo que favorece la implementación de modelos analíticos capaces de transformar información histórica en conocimiento útil para la gestión empresarial. Como resultado, las organizaciones pueden desarrollar estrategias más precisas para responder a las condiciones cambiantes del mercado y fortalecer su competitividad.

7.5 Machine Learning

El machine learning (ML) o aprendizaje automático es un concepto de la inteligencia artificial que analiza los algoritmos y modelos estadísticos que permite a los sistemas optimizar su eficiencia en una labor concreto a través de la experiencia, sin ser programados de manera autónoma para cada

escenario factible (Mitchell, 1997).

En el ámbito de la gestión de inventarios, el ML supervisado ha identificado la aplicación en la predicción de la demanda futura a nivel de SKU (Stock Keeping Unit), en la predicción del lead time de distribuidores y en la detección de órdenes atípicas. El aprendizaje no supervisado, por su parte, se ha empleado a la segmentación de productos conforme a su comportamiento histórico de ventas, a la detección de patrones de demanda con estacionalidad ignota y a la agrupación de clientes basándose a su patrón de consumo (Real Carbonneau, 2008)

Dentro de las variadas especialidades que conforman la inteligencia artificial, el Machine Learning o aprendizaje automático se ha posicionado como una de las tecnologías con mayor prevalencia

La relevancia del aprendizaje automático ha incrementado exponencialmente debido al avance exponencial de los datos creados por las empresas. Actualmente, entidades de diversos ámbitos utilizan esta tecnología para optimizar procesos relacionados con marketing, finanzas, logística, producción y gestión de inventarios (China, 2021)

En el sector de materiales para construcción, el Machine Learning autoriza analizar el comportamiento comercial de los productos, identificar patrones de consumo y desarrollar mecanismos de clasificación que optimicen una gestión más eficiente de los inventarios. Estas capacidades denotan una fortaleza competitiva para las empresas que buscan optimizar el uso de sus recursos y mejorar sus niveles de servicio (Monreal, 2025).

La categorización fundamental del ML distingue varios paradigmas de aprendizaje. El aprendizaje supervisado se desempeña sobre agrupaciones de datos etiquetados, datos de entrada (inputs) y datos de salida (outputs)— y su propósito conectar las variables independientes con las dependientes correspondientes; regresión lineal, árboles de decisión y redes neuronales profundas son

ejemplos representativos (Trevor Hastie , Robert Tibshirani , Jerome Friedman, 2009). El aprendizaje no supervisado opera estrictamente con datos de entrada sin etiquetas y su finalidad es detectar patrones en los datos, el ACP y el K-Means pertenecen a este modelo. (Barto, 2018). Finalmente, el aprendizaje por refuerzo se basa en un proceso de toma de decisiones secuenciales, en el cual un agente interactúa con un entorno y aprende una política óptima $\pi = (a/s)$ que maximiza una recompensa acumulada esperada. A diferencia de los enfoques anteriores, este paradigma se caracteriza por el aprendizaje a partir de la experiencia y la retroalimentación dinámica (Barto, 2018).

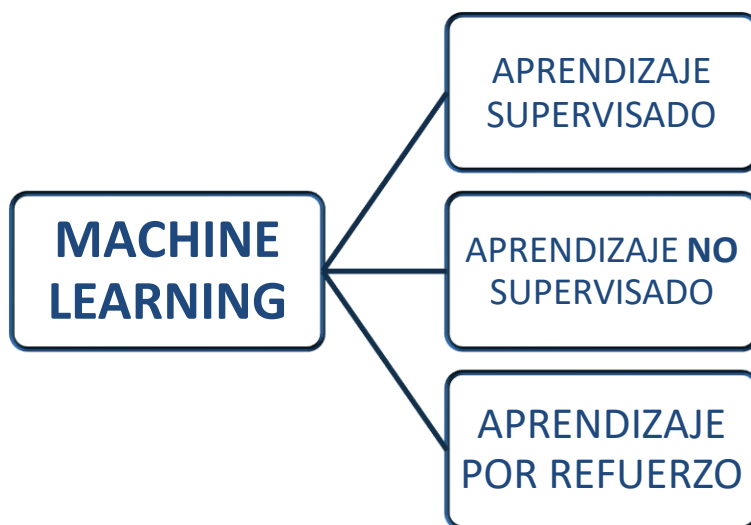


Ilustración 2 Tipos de aprendizajes del Machine Learning. Fuente: Elaboración propia

Dentro de las técnicas de IA, el aprendizaje no supervisado —y especialmente el clustering— abarca un lugar peculiar en la gestión de inventarios porque no implica que los datos estén etiquetados con las categorías de preferencia. En un entorno donde la clasificación vigente del portafolio puede ser injustificada, obsoleta o establecida en estándares únicamente comerciales, el clustering posibilita revivir la segmentación natural de los productos desde de su manejo histórico observado, sin exigir

una estructura predefinida sobre los datos (Jain, 2010).

7.5.1 Aprendizaje no supervisado

El aprendizaje no supervisado es una perspectiva de aprendizaje automático donde el modelo se entrena con un conjunto de datos no etiquetados. Es decir, el aprendizaje no supervisado, es el mismo algoritmo que descubre los patrones mediante los datos sin tener una etiqueta precisa

A diferencia de otros enfoques de aprendizaje automático, este tipo de modelos busca identificar patrones, relaciones y estructuras ocultas dentro de la información disponible sin depender de resultados conocidos durante el proceso de entrenamiento (Chen, 2024).

En el campo de la gestión de empresas, la capacidad de encontrar patrones que no son visibles resulta muy útil en los casos en que las organizaciones no tienen clasificaciones establecidas previamente o donde se requiere la detección de comportamientos que no son evidentes cuando se emplean métodos de análisis más tradicionales. De este modo, se pueden realizar estrategias más precisas y apoyadas en pruebas que se extraen de las propias bases de datos. (Lewotsky, 2022)

Dentro del sector de materiales para construcción, las técnicas de aprendizaje no supervisado han comenzado a utilizarse para analizar el comportamiento comercial de diferentes productos, identificar similitudes entre artículos y generar segmentaciones que faciliten la administración de inventarios. La creciente disponibilidad de datos provenientes de ventas, rotación de productos y movimientos logísticos ha favorecido la implementación de este tipo de herramientas analíticas en procesos relacionados con la planificación y control de recursos.

De esta misma forma, el aprendizaje no supervisado facilita identificar grupos de productos con comportamientos similares en términos de demanda, frecuencia de venta o niveles de rotación, facilitando la creación o elección de estrategias determinadas para cada categoría.

Esta ventaja, es de suma importancia para las organizaciones que dedican sus actividades a la comercialización y/o distribución de una amplia variedad de productos y requieren optimizar la gestión de inventarios, para así mejorar la operatividad y reducir costos asociados al almacenamiento (Purcell, s.f.).

En consecuencia, el aprendizaje sin supervisión es una variable primordial para el análisis exploratorio de los datos empresariales, debido a que permite rastrear patrones que aportan una mayor comprensión de la conducta de los productos, así como también facilita tomar decisiones que van encaminadas a la optimización de los procesos organizacionales.

7.6 Análisis de Componentes Principales

El Análisis de Componentes Principales (ACP) es una técnica de disminución de dimensionalidad que modifica un grupo de variables originales correlacionadas en un nuevo conjunto de variables no correlacionadas —denominadas componentes principales— ordenadas de mayor a menor varianza explicada (Jolliffe, 2002)

(Navarro Céspedes, 2010) menciona que, el análisis de componentes principales (ACP) realiza dos acciones fundamentales: cuantifica las variables originales y disminuye la dimensionalidad de los datos. Si se realiza un análisis correcto, cada variable debe estar bien representada (correlación alta) en una dimensión pobremente representada (correlaciones bajas) en las demás. Si el ACP se realiza sin revisar los supuestos, no se podrá confiar completamente en los resultados. Mediante esta situación un ACP no lineal o categórico con cuantificaciones adecuadas es una alternativa.

De tal manera, el ACP determina la descomposición en valores únicos de la **matriz de covarianza** de los datos, obteniendo vectores propios que determinen nuevas direcciones suministrándoles en el espacio de variables y valores que cuantifican la varianza capturada por cada dirección. (Grané, 2002)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

n : representa el número de observaciones o productos analizados.

p : corresponde al número de variables consideradas para cada producto.

x_{ij} : representa el valor observado de la variable j sobre el individuo i .

A partir de esta matriz, el ACP realiza una transformación lineal con el propósito de obtener un nuevo espacio de variables que conserve la mayor cantidad de información contenida en los datos originales. Esta transformación se expresa mediante la siguiente ecuación:

$$Z = XW$$

Donde:

X : matriz de datos original.

W : representa la matriz de vectores propios (eigenvectors), obtenidos a partir de la matriz de covarianza o correlación.

Z : matriz transformada que contiene los componentes principales.

El ACP encuentra la transformación lineal W tal que $Z = XW$, donde las columnas de Z son ortogonales entre sí y Z_1 aclara la fracción predominante posible de la varianza total, Z_2 expone el mayor índice restante, y así sucesivamente (Jolliffe, 2002).

Cada componente principal demuestra una proporción determinada de la variabilidad total presente de los datos. Esta consideración se decreta a través de los valores propios o más conocidos como “eigenvalues” los cuales se ordenan de manera descendente conforme a la cantidad de

información que representen.

$$(\lambda_1 > \lambda_2 > \dots > \lambda_p)$$

En PCA los autovectores (eigenvectors) se usan para construir los componentes principales, los cuales son vectores que se emplean para simbolizar la mayor cantidad de variabilidad en el conjunto de datos.

Por lo tanto, el número de elementos a mantener se define usualmente por el criterio de varianza acumulada explicada —entre el 70% y el 90% de la varianza total— o por el análisis del scree plot.

El ACP no es solo un método de preprocesamiento, sino también una herramienta exploratoria que concede visualizar la estructura del portafolio de productos antes de aplicar el clustering (Williams, 2010). La representación de los productos en el plano definido por los dos primeros componentes principales —biplot— muestra agrupaciones naturales, productos atípicos y relaciones entre variables, lo cual permite una elección del número óptimo de clústeres en el K-Means.

Es por ello, que el ACP es de gran utilidad para sintetizar información proveniente de múltiples variables, porque permite identificar factores con mayor relevancia para poder explicar la variabilidad de los datos. Para poder facilitar la interpretación de fenómenos complejos y permite una mejor comprensión de las diferentes relaciones que pueden existir entre las variables utilizadas. Para ello, es de suma importancia confirmar que los datos cumplan los supuestos requeridos para implementar esta técnica ya que al no cumplirse la solidez de los resultados puede verse afectada. En el caso de variables categóricas, el uso del ACP categórico representa una alternativa adecuada para obtener análisis más precisos y confiables.

7.7 K-Means

Algoritmo K-Means (MacQueen, 1967) es uno de los más simples y conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (fijados a priori). La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano.

El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente. El problema del empleo de estos esquemas es que fallan cuando los puntos de un grupo están muy cerca del centroide de otro grupo, también cuando los grupos tienen diferentes tamaños y formas.

El algoritmo K-Means es el método de clustering por partición más ampliamente utilizado en aprendizaje automático (Jain, 2010). Su objetivo es particionar un conjunto de n observaciones en k grupos mutuamente excluyentes y exhaustivos, de modo que cada observación pertenezca al cluster cuyo centroide (media vectorial del grupo) minimice su distancia euclidiana.

El algoritmo mantiene un proceso iterativo que se basa en dos pasos, denominados como el algoritmo de Lloyd (Lloyd, 1982). El primer paso (paso de atribución), cada dato se asigna al clúster cuyo centroide minimiza su distancia euclidiana. El segundo paso (paso de actualización) los centroides se recalculan como la media de todas las observaciones que fueron asignadas a cada clúster. Este proceso se repite hasta la convergencia (cuando las asignaciones no cambian entre iteraciones) o hasta alcanzar un número máximo de iteraciones predefinido. La convergencia está garantizada matemáticamente, aunque puede corresponder a un mínimo local y no al global (MacQueen, 1967).

El objetivo del algoritmo consiste en minimizar la variabilidad existente dentro de cada grupo mediante la siguiente función:

$$J = \sum_{i=1}^k \sum_{x \in C_i} ||x - \mu_i ||^2$$

J : función objetivo o inercia.

k : número de clústeres.

C_i : conjunto de observaciones pertenecientes al clúster i .

x : observación.

μ_i : centroide del clúster i .

La dependencia de la inicialización aleatoria de los centroides es la principal vulnerabilidad del K-Means básico. (David Arthur, 2007) propusieron K-Means, que selecciona los centroides iniciales de manera probabilística favoreciendo puntos separados entre sí, incrementando significativamente la probabilidad de convergencia al óptimo global y reduciendo la varianza entre ejecuciones

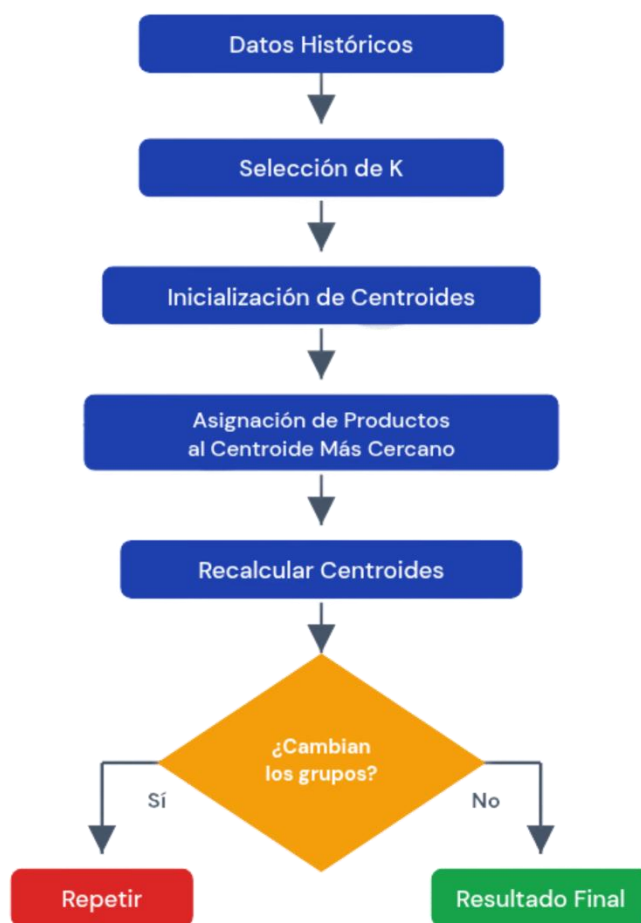
La determinación del número óptimo de clústeres k es el desafío práctico central del K-Means. En la literatura se documentan múltiples criterios complementarios. El método del codo (elbow method) grafica la inercia en función de k e identifica el punto donde la reducción marginal de inercia comienza a decrecer de manera significativa (Thorndike, 1953).

El coeficiente de silueta mide para cada una de las observaciones, la cohesión interna de su clúster y su separación respecto a los clústeres cercanos (clúster vecino), con valores cercanos a 1 indicando una asignación adecuada y valores negativos que indican una posible clasificación errónea (Rousseeuw, 1986).

El índice de Calinski-Harabasz mide la relación existente entre la dispersión interclúster e

intracluster, con valores más altos que indiquen clusters más compactos y correctamente separados (Harabasz, 1974).

Ilustración 3 Proceso de agrupamiento mediante el algoritmo K-Means, desde la selección del número de clústeres hasta la obtención del resultado final. Fuente: Elaboración propia



7.7.1 Aplicación de K-Means en la Optimización de Stock

El uso del algoritmo K-Means en la optimización de inventarios es un ejemplo bien documentado en la literatura científica más reciente, como alternativa y complemento a los métodos clásicos de segmentación de portafolios. El principio de esta metodología se basa en que productos con comportamientos de demanda similares deben ser tratados con políticas de reposición similares y que el uso de K-Means es la herramienta más adecuada para detectar esos grupos de similitudes de una forma objetiva, replicable y escalable.

(Lolli, 2013) realizaron una evaluación sistemática de la técnica de clustering como forma de clasificar inventarios en un contexto multidimensional comparando K-Means, la clasificación jerárquica y la clasificación ABC convencional. Los resultados mostraron que K-Means genera clústeres más estables y de mayor homogeneidad interna que la clasificación jerárquica para portafolios de tamaño mediano y grande, y afirmaron que la clasificación multidimensional obtenida se mostró en la predicción mejor que el ABC a la hora de caracterizar el comportamiento de los productos, siempre que las variables que entran en la clasificación sean representativas de la realidad del inventario.

(Huber, 2017) destacan que la fase de caracterización e interpretación de los clústeres es tan importante como la fase técnica de su generación. Un clúster cuya descripción estadística no puede ser traducida en una política de reposición diferenciada carece de valor práctico. En la optimización de stock, la interpretación de los clústeres debe producir recomendaciones concretas: por ejemplo, un clúster de productos de alta demanda y baja variabilidad puede gestionarse con EOQ estándar y bajo stock de seguridad; un clúster de alta variabilidad y lead time largo requiere stock de seguridad elevado y modelos de reposición probabilísticos; un clúster de productos de demanda intermitente puede beneficiarse de modelos de pronóstico especializados como el método de Croston.

Tabla 1: Matriz de conceptos y aplicaciones en Inteligencia Artificial y métodos clustering. Fuente: Elaboración propia.

CONCEPTO	DEFINICIÓN	APLICACIÓN
Inteligencia Artificial	Sistemas que simulan inteligencia humana	Automatización
Machine Learning	Rama de la IA que aprende de datos	Predicciones y análisis
Aprendizaje NO supervisado	Identifica patrones sin etiquetas	Segmentación de productos
K-means	Algoritmo de clustering	Agrupación de inventario

Desde el punto de vista de los resultados esperados, la aplicación de políticas de reposición diferenciadas por clúster causa impactos financieros y operativos medibles. (Carbonneau, 2011) menciona que se reportan mejoras en el nivel de servicio (medido como porcentaje de demanda satisfecha sin quiebre de stock) de hasta un 8% en escenarios comparables, mientras que (Silver, 2016) documentan que la reducción del stock de seguridad basada en una mejor segmentación del portafolio puede representar entre el 10% y el 30% del capital inmovilizado en inventario. En empresas de distribución de consumo masivo con portafolios amplios, estos porcentajes se traducen en cifras de liberación de capital de trabajo y reducción de costos de almacenaje que justifican con amplitud la inversión en el desarrollo del modelo analítico.

(Domingos, 2012) advierte que la validación de un modelo de ML no puede descansar únicamente en métricas estadísticas internas como la inercia o el coeficiente de silueta; la prueba definitiva de su utilidad es su capacidad para generar mejores decisiones de negocio que las alternativas disponibles. Para la optimización del stock, esto implica que la validación del modelo K-Means debe incluir una comparación retrospectiva: simular qué habría ocurrido con los niveles de inventario, los quiebres de stock y los costos de almacenaje si las políticas de reposición resultantes del modelo hubieran sido aplicadas en el período histórico de análisis, y comparar ese escenario hipotético con los resultados reales observados durante el mismo período.

8 MARCO CONCEPTUAL

8.1 Inteligencia Artificial

La definición de IA mantiene múltiples percepciones dependiendo el concepto en el que se lo aplica (Stuart Russell, 2008) definen a la inteligencia artificial como diseño y la construcción de agentes inteligentes que reciben percepciones del entorno y emprenden acciones que afectan ese entorno. En otras palabras, comprenden a la inteligencia artificial no tanto como una actividad humana, sino como un sistema que simula la inteligencia natural.

La inteligencia artificial es una ciencia que tiene como objetivo el diseño y construcción de máquinas capaces de imitar el comportamiento inteligente de las personas (Munárriz, 1994)

Dentro del campo de la Inteligencia Artificial, el Machine Learning (aprendizaje automático) se define como un enfoque que permite a los sistemas aprender patrones y relaciones directamente a partir de los datos, sin necesidad de una programación explícita basada en reglas. Este paradigma resulta especialmente relevante en contextos caracterizados por grandes volúmenes de información y alta variabilidad, donde los enfoques tradicionales resultan limitados (Mitchell, 1997).

8.1.1 Machine Learning

(Foster Provost, 2013) señalan que la calidad del efecto de cualquier modelo de machine learning está sujeta por la calidad de los datos usados en su entrenamiento. Para la optimización de inventarios, esto conlleva que los archivos históricos de ventas, ingresos de mercancías, ajustes de inventario y devoluciones están obligados a estar adecuadamente capturados y saneados antes de ser empleados como insumo del modelo.

8.1.2 Modelo no supervisado

La finalidad más relevante que tiene el aprendizaje no supervisado es la de encontrar tipos de comportamientos que tienen similitudes en los datos analizados, lo que permite constituir agrupaciones naturales que facilitan la visión de grandes cantidades de datos. A través de las características comunes, estos modelos ayudan a transformar datos dispersos en conocimiento que favorezca la toma de decisiones empresariales. (Rosicart, 2026)

Por su parte, el aprendizaje no supervisado opera sobre conjuntos de datos que no incluyen etiquetas de salida, centrándose en la identificación de estructuras latentes, patrones o agrupaciones dentro de los datos. En este caso, el objetivo no es predecir una variable específica, sino modelar la distribución subyacente de las observaciones, lo que resulta particularmente útil en análisis exploratorios, reducción de dimensionalidad y segmentación de datos complejos (Hastie, Tibshirani, & Friedman, 2009).

8.2 Análisis de Componentes Principales

A nivel matemático, el Análisis de Componentes Principales especificar la descomposición en valores pertenecientes de la matriz de covarianza de los datos, logrando vectores propios que establezcan nuevas direcciones proporcionándoles en el espacio de variables y valores propios que cuantifican la varianza capturada por cada dirección.

El número de elementos que se deben mantener se definirá usualmente por el criterio de varianza acumulada explicada —entre el 70% y el 90% de la varianza total— o por el análisis del scree plot.

El ACP no es solo un método de preprocesamiento, sino también una herramienta exploratoria que concede visualizar la estructura del portafolio de productos antes de aplicar el clustering

(Williams, 2010). La representación de los productos en el plano definido por los dos primeros componentes principales —biplot— muestra agrupaciones naturales, productos atípicos y relaciones entre variables, lo cual permite una elección del número óptimo de clústeres en el K-Means.

Sin embargo, es una técnica estadística multivariante muy utilizada en trabajos de análisis de datos y en el aprendizaje de máquinas. Su principal finalidad es reducir la dimensionalidad de un conjunto de datos manteniendo la mayor cantidad de información posible que hay en las variables originales (Jaadi, s.f.).

(Navarro Céspedes, 2010) menciona que,

El análisis de componentes principales (ACP) realiza dos acciones fundamentales: cuantifica las variables originales y disminuye la dimensionalidad de los datos. Si se realiza un análisis correcto, cada variable debe estar bien representada (correlación alta) en una dimensión pobremente representada (correlaciones bajas) en las demás. Si el ACP se realiza sin revisar los supuestos, no se podrá confiar completamente en los resultados. Mediante esta situación un ACP no lineal o categórico con cuantificaciones adecuadas es una alternativa.

Es por ello, que el ACP es de gran utilidad para sintetizar información proveniente de múltiples variables, porque permite identificar factores con mayor relevancia para poder explicar la variabilidad de los datos. Para poder facilitar la interpretación de fenómenos complejos y permite una mejor comprensión de las diferentes relaciones que pueden existir entre las variables utilizadas. Para ello, es de suma importancia confirmar que los datos cumplan los supuestos requeridos para implementar esta técnica ya que al no cumplirse la solidez de los resultados puede verse afectada

En el caso de variables categóricas, el uso del ACP categórico representa una alternativa

adecuada para obtener análisis más precisos y confiables.

8.3 K-Means

El algoritmo K-Means es el método de clustering por partición mayormente utilizado en aprendizaje automático (Jain, 2010). Su objetivo es dividir un conjunto de n observaciones en k grupos mutuamente excluyentes y exhaustivos, de modo que cada observación pertenezca al clúster cuyo centroide (media vectorial del grupo) minimice su distancia euclidiana. La función objetivo que minimiza es la inercia o suma de distancias cuadráticas intracluster: $J = \sum_i \sum_{x \in C_i} ||x - \mu_i||^2$ donde μ_i es el centroide del clúster i .

Este algoritmo conserva una metodología interactiva de dos pasos que es denominado como el algoritmo de Lloyd (Lloyd, 1982). En el punto de asignación, cada observación se asignará al clúster cuyo centroide disminuye su distancia euclidiana.

Ahora, en el paso de actualización, los centroides se vuelven a procesar como la media de todas las observaciones asignadas a cada clúster. El proceso se repite hasta convergencia (cuando las asignaciones no cambian entre iteraciones) o hasta llegar a un número límite de iteraciones ya establecido. La convergencia está garantizada de forma matemática, aunque puede corresponder a un mínimo local y no al global (MacQueen, 1967).

La dependencia de la inicialización aleatoria de los centroides es la principal vulnerabilidad del K-Means básico. (David Arthur, 2007) propusieron K-Means, que selecciona los centroides iniciales de manera probabilística favoreciendo puntos separados entre sí, incrementando significativamente la probabilidad de convergencia al óptimo global y reduciendo la varianza entre ejecuciones.

La determinación del número óptimo de clústeres k es el desafío práctico central del K-Means.

En la literatura se registran múltiples criterios que se complementan. El método del codo (elbow method) grafica la inercia en función de k y reconoce el punto donde la reducción marginal de inercia comienza a decrecer de manera significativa (Thorndike, 1953).

El coeficiente de silueta medida, para cada observación, la cohesión interna de su clúster y su separación respecto a los clústeres vecinos, con valores cercanos a 1 indicando una asignación adecuada y valores negativos indicando posible clasificación errónea (Rousseeuw, 1986).

El índice de Calinski-Harabasz evalúa la relación entre la dispersión Inter clúster e intracluster, con valores más altos indicando clústeres más compactos y bien separados (Harabasz, 1974).

8.3.1 Clustering

Se conoce como clustering a un algoritmo del Machine Learning que ordena y categoriza varios datos u observaciones en grupos o clústeres basados en similitudes o patrones. Este algoritmo se puede utilizar de varias formas, tanto para explorar un conjunto de datos como para monitorear procesos en curso. También se suele utilizar para análisis exploratorio de datos con un nuevo grupo de datos para analizar patrones, tendencias y también valores atípicos.

Es provechoso considerar al clustering como un intento de encontrar agrupaciones naturales en los datos para ver qué categorías pueden existir y qué define esas categorías. Los clústeres pueden ayudarle a encontrar relaciones subyacentes entre puntos de datos para ver qué características o rasgos se comparten entre las categorías (Jaadi, s.f.).

9 MARCO LEGAL

El presente trabajo de investigación se desarrolla a partir del análisis de información relacionada con la comercialización y gestión de inventarios de productos para acabados de construcción en una empresa ecuatoriana. Los datos por utilizar contienen información de registros comerciales y operativos, como historial de ventas, nivel de inventario y rotación de este. Aunque el estudio no trata datos personales de clientes o empleados, sí usa información a nivel empresarial. Si se divulga esa información sin permiso, la competitividad y los intereses comerciales de la organización podrían verse afectados.

Por eso, se toman medidas para asegurar un manejo ético, responsable y seguro de la información que se usa durante el desarrollo de la investigación. Además, los datos se usarán solo con fines académicos y para análisis estadístico. En todo momento se cuidará la confidencialidad de la empresa y se evitará mostrar información estratégica o sensible que pueda afectar sus operaciones o su desempeño en el mercado.

Bajo este contexto tomamos como principal referencia la Ley Orgánica de Protección de Datos Personales (LOPD), promulgada en 2021, que constituye la normativa nacional aplicable en materia de protección de datos en el Ecuador. En base a lo establecido en el Artículo 2 de dicha ley, quedan excluidos de su ámbito de aplicación los datos anonimizados o aquellos que no permiten identificar directa o indirectamente a una persona natural. En concordancia con esta disposición, los datos empleados en la presente investigación han sido tratados de forma agregada y sin elementos que permitan la identificación de personas naturales.

a) Juridicidad: El tratamiento de la información otorgada por la empresa se desarrollará en base al marco normativo vigente, garantizando que el uso de los datos sea lícito y coherente con los fines científicos y académicos de la investigación (Ley Orgánica De Protección De Datos Personales, 2021)

b) Lealtad: Los datos facilitados serán gestionados de forma ética y responsable, asegurando una relación basada en la confianza con la empresa colaboradora y evitando prácticas que puedan generar interpretaciones indebidas o usos distintos a los previamente acordados (Ley Orgánica De Protección De Datos Personales, 2021)

c) Transparencia: La entidad que provee la información contará con conocimiento claro y oportuno sobre los objetivos del tratamiento de los datos, del proceso de investigación y el alcance de su utilización durante el desarrollo del modelo de clasificación. (Ley Orgánica De Protección De Datos Personales, 2021)

d) Finalidad: La información recopilada será utilizada únicamente para el diseño, entrenamiento y validación del algoritmo orientado a la optimización del stock, sin ser empleada para fines ajenos a los objetivos del estudio (Ley Orgánica De Protección De Datos Personales, 2021).

e) Pertinencia y minimización: Se limitará la recopilación de datos a aquellos estrictamente necesarios para el funcionamiento y desempeño del modelo, evitando la inclusión de variables irrelevantes o que no aporten valor analítico al proceso de clasificación (Ley Orgánica De Protección De Datos Personales, 2021)

f) Proporcionalidad: El volumen y tipo de información tratada se ajustará de manera proporcional al alcance y necesidades del proyecto, procurando no procesar datos que excedan los requerimientos técnicos y metodológicos de la investigación (Ley Orgánica De Protección De Datos Personales, 2021)

g) Confidencialidad: Los datos suministrados serán considerados de carácter reservado y se aplicarán medidas que impidan su acceso, uso o divulgación no autorizada, reconociendo su valor estratégico y sensible para la empresa que los proporciona (Ley Orgánica De Protección De Datos Personales, 2021)

h) Calidad y exactitud: La investigación se apoyará en información depurada, consistente y

actualizada, dado que la calidad de los datos constituye un factor determinante para la fiabilidad de los resultados obtenidos mediante el modelo de clasificación (Ley Orgánica De Protección De Datos Personales, 2021).

10 METODOLOGÍA

Este trabajo de investigación toma un enfoque cuantitativo, con un alcance descriptivo-exploratorio, orientado a la segmentación de productos de inventario mediante modelos de aprendizaje no supervisado. Esta metodología combina dos algoritmos que se complementan: Análisis de componentes principales (ACP), que se emplea con la finalidad de reducir la dimensionalidad y de eliminar la colinealidad entre variables, y el algoritmo K-means que busca agrupar los datos sobre el espacio resultante de los componentes principales con la finalidad de segmentar los productos del inventario según comportamientos similares, con el propósito de generar información relevante para la gestión de existencias y la toma de decisiones logísticas.

10.1 Fuente de datos

La base de datos corresponde a un kardex general de inventario (registro histórico de entradas, salidas y saldos por artículo) correspondiente al periodo 2023–2025 de la empresa TOALCO SAS. La base de datos contiene, para cada movimiento, la fecha de la transacción, el producto, la línea o categoría de materiales, el saldo anterior, las cantidades movidas y el costo asociado.

10.2 Tratamiento de los datos

Se inicia realizando una **ingeniería de características** con la finalidad de crear nuevas variables que describan el comportamiento de cada producto en términos de demanda, inventario y costos. La finalidad es transformar los datos transaccionales en métricas agregadas y significativas para el clustering.

10.3 Preparación Inicial de Datos

Se inicia creando una copia del DataFrame y se suprimen las filas donde las columnas 'producto' o 'fecha' tienen valores nulos. Esto nos va a asegurar que solo se procesen registros completos y válidos.

Se extrae el mes y año de la columna 'fecha' y se guarda como un período ('M') en una nueva columna 'mes'. Esto es indispensable para las agregaciones mensuales.

Las columnas que contienen valores numéricos se convierten explícitamente a tipo numérico y se reemplaza cualquier valor que no pueda ser convertido a número con NaN (Not a Number).

10.3.1 Creación del Maestro de Productos (máster)

Se crea un Data Frame master que contiene información básica y única para cada producto, como su nombre (Artículo) y su línea (Materiales). Se agrupa por 'producto' para asegurar la unicidad y se toma el primer valor encontrado para 'Artículo' y 'Línea'.

10.4 Variables de Demanda

10.4.1 Demanda Total

En el ámbito de la gestión logística y la administración avanzada de la cadena de suministro, la demanda total se erige como el parámetro empírico fundacional para comprender el comportamiento estructural del mercado.

Para formalizar este concepto en el modelo matemático, la ecuación de la demanda total se expresa como la sumatoria de las salidas individuales. La fórmula se define de la siguiente manera:

En esta expresión matemática, cada componente cumple un rol específico para la agregación del flujo.

$$Dp = \sum (t = 1 \text{ hasta } T) qp, t$$

Dónde:

Dp : representa la demanda agregada total del producto específico 'p' a lo largo de todo el periodo de estudio.

(Σ) : indica que se debe realizar una adición acumulativa desde el momento temporal inicial 't = 1' hasta llegar al límite superior

T : el cual corresponde al horizonte temporal total evaluado (que puede estar medido en días, semanas o meses, dependiendo de la granularidad de los datos extraídos del sistema de información)

qp, t : es el núcleo operativo de la ecuación, ya que define la cantidad física o volumétrica que fue efectivamente despachada, facturada o consumida del producto 'p' durante ese intervalo de tiempo discreto 't'.

Esta sumatoria lineal garantiza que ninguna transacción, por mínima que sea, se omita en el cálculo del peso comercial del artículo.

La correcta aplicación y entendimiento de esta expresión matemática permite a los algoritmos de agrupamiento no supervisado, como K-means, realizar una diferenciación con rigor matemático. Al alimentar el algoritmo con el vector de demanda total, el sistema estadístico logra discriminar con alta precisión entre aquellos artículos que fungen como pilares de la operación comercial —es decir, los de salida masiva y constante— frente a aquellos cuya salida es marginal, esporádica o puramente residual.

Esta diferenciación volumétrica facilita enormemente a los analistas la toma de decisiones estratégicas, permitiéndoles diseñar políticas de reabastecimiento que mitiguen problemas crónicos de sobre stock en productos de baja salida, o de desabastecimiento en productos estrella. En entornos

comerciales altamente competitivos como el de los materiales de construcción, hay que asegurar que la métrica de demanda total esté matemáticamente depurada y exenta de sesgos transaccionales es el primer paso vital para garantizar el éxito de cualquier modelo predictivo o de segmentación automatizada de inventarios.

10.4.1.1 Suma acumulada del consumo del producto i en todo el horizonte histórico.

Estas variables describen el patrón de consumo o salida de cada producto. Se agrega la cantidad de salidas por producto y por mes. Se transforman los datos mensuales a un formato de tabla pivote, donde las filas son producto y las columnas son los meses. Se usa *reindex* para asegurar que todos los meses del rango completo estén presentes para cada producto, rellenando con 0 cuando no hubo ventas en un mes específico. Esto es crucial para calcular métricas como la intermitencia.

10.4.2 Demanda

Suma de todas las cantidades vendidas (o salidas) de un producto a lo largo de todo el período de tiempo disponible.

$$Dp = \sum (t = 1 \text{ hasta } T) qp, t$$

10.4.2.1 Media Mensual

Promedio de las cantidades vendidas (o salidas) de un producto por mes a lo largo del período.

$$\bar{X}_m = \frac{\sum_i^{n_m} x_{i_m}}{n_m}$$

Donde:

$\bar{x}_m$ el promedio de la variable por mes (media aritmética)

x_{i_m} valor de cada observación i que pertenece al mes m

$\sum_i^n x_i$ la suma de todos los valores

n_m el número total de datos (observaciones)

10.4.2.2 Desviación Estándar Mensual (*std_mensual*)

Medida de la dispersión o variabilidad de las cantidades vendidas (o salidas) de un producto alrededor de su media mensual.

$$s = \frac{\sum_1^N (x_i - \bar{X})^2}{n - 1}$$

Dónde

x_i : cada valor individual de la variable

$\bar{x}$: la media (promedio) de esa variable

$(x_i - \bar{x})^2$: la distancia de cada valor respecto a la media, elevada al cuadrado

n : número total de observaciones

$n - 1$: se usa “ $n - 1$ ” cuando trabajas con una muestra en vez de toda la población, para corregir el sesgo del estimado

Coefficiente de Variación (CV) de la Demanda

Indica la variabilidad relativa de la demanda. Un CV alto sugiere demanda irregular, mientras que un CV bajo indica demanda más estable. Se manejan divisiones por cero y se rellenan los NaN resultantes con 0.

$$CV = \sigma / \mu$$

Dónde

CV : coeficiente de Variación de la demanda del producto.

σ : desviación estándar muestral o poblacional de los despachos históricos, la cual mide la dispersión absoluta de los datos respecto al centro.

μ : media aritmética de dicha demanda en el mismo periodo.

Al dividir la desviación estándar (' σ ') entre la media (' μ '), la ecuación elimina la influencia de las unidades de medida y la magnitud bruta del producto, entregando un número relativo puro. Esto significa que el coeficiente de la fórmula mide directamente qué tan grande es la volatilidad de la demanda como una fracción del tamaño promedio de las ventas de ese SKU.

10.4.3 Intermitencia

Mide la proporción de meses en los que un producto no tuvo demanda. Un valor cercano a 1 indica alta intermitencia (demanda esporádica), mientras que un valor cercano a 0 indica demanda continua.

$$I = \frac{n_o}{n}$$

Dónde

I : intermitencia (entre 0 y 1, o se multiplica por 100 para tenerla en %)

n_o : número de periodos (meses) **sin** ninguna venta del producto

n : número total de periodos analizados

Frecuencia de Salidas (frecuencia)

Número de registros (eventos) en los que un producto tuvo una salida (cantidad 0.148 mayor que cero). Mide cuántas veces un producto ha sido registrado con alguna venta o movimiento.

$$F = \sum_{t=1}^n 1(Q_t > 0)$$

Dónde

F : frecuencia de salidas (número de veces que hubo una salida)

n : número total de periodos analizados (por ejemplo, días o meses)

Q_t : cantidad vendida/despachada en el periodo t $1(Q_t > 0)$ función indicadora que vale 1 si en ese periodo hubo salida ($Q_t > 0$), y **0** si no hubo ninguna

Variables de Saldo y Rotación de Inventario

Estas variables caracterizan la gestión del inventario de cada producto.

Saldo Promedio

Promedio de los saldos anteriores registrados para un producto.

Porcentaje de Saldo

Indica la frecuencia con la que el saldo de inventario de un producto ha sido negativo, lo que puede ser una señal de problemas de stock o de registro.

Rotación de Inventario (rotación)

Mide cuántas veces el inventario promedio de un producto es vendido o reemplazado durante un período. Un valor alto indica que el inventario se vende rápidamente. Se manejan divisiones por y se rellenan los NaN resultantes con 0.

El índice de rotación de inventarios constituye el termómetro empírico, operativo y financiero por excelencia para medir la eficiencia de las políticas de gestión logística implementadas por cualquier entidad comercial. Esta métrica analiza directamente la velocidad con la que la empresa es capaz de convertir sus existencias en ventas concretas, reflejando el dinamismo del mercado y el asertividad del departamento de compras. En otras palabras, esta medida representa la velocidad del capital de trabajo: los valores altos reflejan un flujo de caja óptimo, buena aceptación en el mercado y menor riesgo de productos caducados o depreciación del inventario. Por otro lado, los valores bajos demuestran ineficiencias graves y crónicas de sobrestock, revelando proyecciones de compras incorrectas que castigan y congelan la liquidez financiera de la organización. Por estas razones, la rotación es una variable descriptiva irremplazable.

La expresión matemática que define la frecuencia con la que el stock total es renovado compara el flujo saliente con el nivel de existencias, estructurándose así:

$$R \text{ Rotación} = \frac{Dp}{S} \text{ promedio}$$

En esta ecuación, el término 'Rotación' indica el índice numérico de veces que el almacén ha vaciado y vuelto a llenar su inventario de un artículo determinado en el tiempo analizado.

Dónde:

Dp: Demanda Total acumulada en el periodo de estudio, la cual simboliza la cantidad total de bienes que han fluido fuera de la empresa hacia los clientes.

S: (o saldo promedio) corresponde al nivel de existencias físicas o financieras mantenidas en el almacén a lo largo del mismo horizonte temporal.

Este saldo promedio se calcula típicamente sumando el inventario inicial y el inventario final del periodo y dividiéndolo por dos, aunque en modelos de ERP más avanzados se utiliza el promedio aritmético diario. La relación que establece la fórmula dicta cuántas veces la Demanda (Dp) contiene al Stock Promedio, revelando la tasa de eficiencia logística real del producto.

Variables de Costo Unitario (Cascada de Validez)

Esta sección calcula un costo unitario representativo para cada producto utilizando una lógica de prioridad o 'cascada'.

Costo Promedio, costo último, precio, porcentaje de costo, costo, valor consumido.

Consolidación de Características

Todas las variables calculadas (`demanda_total`, `frecuencia`, `cv_demanda`, `intermitencia`, `costo_unit_prom`, `valor_consumido`, `saldo_promedio`, `rotacion_inventario`, `pct_costo_valido`, `pct_saldo_negativo`) se combinan en un solo DataFrame llamado `feat`. Luego se unen con el DataFrame máster para incluir el 'Articulo' y la 'Linea' de cada producto.

Finalmente, se limpian valores infinitos reemplazándolos con NaN, y luego se eliminan las filas que aún tienen valores NaN, ya que un costo unitario es esencial para el análisis.

10.5 Análisis de Componentes Principales (ACP)

10.5.1 Estandarización Z (StandardScaler)

La etapa metodológica de preprocesamiento de datos representa un eslabón técnico ineludible e imperativo en el campo de la ciencia de datos y el aprendizaje automático para garantizar la validez matemática y la robustez del modelo resultante. Las variables logísticas que alimentan el sistema, por su propia naturaleza descriptiva, operan intrínsecamente en unidades de medida y magnitudes numéricas radicalmente distintas. Por ejemplo, la variable que representa el valor consumido económico de un producto puede ascender fácilmente a decenas de miles o cientos de miles de dólares, mientras que la característica que mide la frecuencia mensual de despachos puede limitarse a dígitos simples o decimales que no superan la decena. Los algoritmos que fundamentan su lógica operativa en el cálculo de distancias y proximidades geométricas dentro de espacios euclidianos, como es el

caso del Análisis de Componentes Principales (ACP) y el algoritmo K-means, sufren de una extrema vulnerabilidad a las disparidades de escala.

Para resolver esta distorsión y equilibrar el peso de las características, se aplica la estandarización estadística mediante la transformación Z, formalizada a través de la siguiente fórmula:

$$z = (x - \mu) / \sigma$$

Donde:

z: puntuación z

x: valor que se está evaluando

μ : media

σ : desviación estándar

El análisis detallado de los componentes de esta fórmula revela el proceso de normalización:

Dónde:

Z: variable resultante o el valor estandarizado (también conocido mundialmente como Z-score), el cual será la nueva coordenada que el modelo procesará.

La variable “x” constituye el valor empírico original de la observación extraída del ERP (el costo o la demanda cruda de un producto).

μ : (mu) media aritmética calculada de esa característica específica en la totalidad de la muestra poblacional, representando el centro de gravedad de los datos.

σ : (sigma) desviación estándar global de la característica, actuando como un factor de escala.

Al restar la media “ μ ” a la observación original “ x ”, el sistema centra la distribución completamente en cero. Posteriormente, al dividir ese resultado por la desviación estándar “ σ ”, la fórmula comprime o extiende los datos para que la varianza resultante sea igual a uno.

Si la matriz de información original no se sujeta a este minucioso proceso de estandarización Z antes de su ingesta algorítmica, aquellas variables con magnitudes numéricas superiores y rangos extensos dominarán de manera desproporcionada la función de costo geométrico. Esto ocasionaría un error fatal en el modelado, anulando por completo la influencia estadística de atributos que, aunque poseen valores numéricos minúsculos (como una tasa de intermitencia del 0.8), son teóricamente igual de críticos para la estrategia de abastecimiento. La aplicación sistemática de esta fórmula sobre cada columna del conjunto de datos asegura una absoluta equidad analítica. Al transformar todas las variables logísticas dispares a un lenguaje numérico universal de desviaciones estándar, el sistema K-means es capaz de trazar distancias justas y descubrir los clústeres subyacentes basándose exclusivamente en el comportamiento genuino de los artículos y no en la unidad métrica que los describe.

10.5.2 Eigenvectores y Eigenvalores (Análisis de Componentes Principales)

El Análisis de Componentes Principales (ACP) es una técnica estadística multivariante de vanguardia orientada fundamentalmente a la resolución analítica del denominado 'problema de la dimensionalidad'. En bases de datos corporativas con múltiples indicadores logísticos y financieros, la abundancia de variables suele generar un exceso de información correlacionada y ruido estadístico que satura el procesamiento computacional de los modelos de Machine Learning. El propósito central de esta técnica es destilar la esencia del comportamiento de los datos, proyectando los miles de observaciones desde su espacio original de alta dimensión hacia un hiperplano geométrico nuevo, mucho más reducido, conformado por dimensiones sintéticas que son completamente ortogonales e

independientes entre sí. Esto facilita una visualización clara del portafolio, previene la multicolinealidad y purifica la matriz de datos para que el algoritmo de agrupamiento K-means funcione a su máxima capacidad y sin redundancias.

Desde una perspectiva matemática rigurosa, este método opera mediante la descomposición espectral de la matriz de covarianza, lo cual se sintetiza en el siguiente sistema de ecuaciones algebraicas:

$$Z = XW, \text{ sujeto a } \lambda_1 > \lambda_2 > \dots > \lambda_p$$

Esta estructura matemática describe la transformación del espacio tridimensional o multidimensional.

Dónde

Z: la nueva matriz de datos transformada, es decir, el conjunto final de componentes principales que serán utilizados por el modelo

X: corresponde a la matriz estandarizada de datos que contiene todas las dimensiones logísticas originales del inventario

W: matriz de peso ortogonal compuesta por los vectores propios (conocidos en álgebra lineal como Eigenvectores); estos vectores actúan como flechas direccionales que indican la inclinación exacta de los nuevos ejes en el hiperplano para lograr maximizar la separación de los datos.

λ : (lambda) define a los valores propios (o Eigenvalores) asociados a cada vector.

Estos Eigenvalores son vitales porque cuantifican con absoluta precisión matemática la magnitud o porcentaje de la varianza total explicada que cada nuevo componente principal ('Z') ha logrado retener del sistema original, justificando su importancia.

Se estipula que los componentes principales extraídos se ordenen de manera estrictamente descendente, basándose en la magnitud de sus Eigenvalores, tal como lo expresa la inecuación ($\lambda_1 > \lambda_2 > \dots > \lambda_p$). Esto garantiza que el primer componente analítico abarque la mayor cantidad de

información posible sobre el inventario, el segundo la siguiente mayor cantidad, y así sucesivamente. Posteriormente, se aplica el criterio empírico de Kaiser para la selección final de las variables, el cual dicta que se deben conservar únicamente aquellos componentes cuyo valor propio asociado (Eigenvalor) sea estrictamente mayor a la unidad matemática. Descartar los vectores con Eigenvalores menores a 1 ayuda a eliminar el ruido de fondo, minimizando la matriz y optimizando de manera sustancial la eficiencia y la velocidad del procesamiento computacional, garantizando que el algoritmo K-means construya clústeres sobre datos sólidos.

Varianza y Covarianza: PCA busca las direcciones (vectores) en el espacio de características donde la varianza de los datos es máxima. Para ello, calcula la matriz de covarianza de las características estandarizadas.

10.5.3 Covarianza

En los ecosistemas logísticos contemporáneos, las variables operativas interactúan constantemente de forma simultánea. Para cuantificar con exactitud la dirección y magnitud de esta variación conjunta entre dos métricas distintas, la estadística introduce el concepto de la covarianza, clave para detectar redundancias en la información.

Calcular la covarianza entre indicadores, como el precio promedio y el volumen de compra, expondrá relaciones conjuntas o contrario. Estos resultados llenarán la matriz de covarianza, entrenando al algoritmo para que agrupe variables correlacionadas, evitando el sobreajuste al perfilar la demanda.

10.6 Ortogonalidad de los Componentes Principales

El principio de ortogonalidad es una condición importante en el Análisis de Componentes Principales ya que garantiza la total independencia estadística entre las recientes dimensiones formadas. Al proyectar datos logísticos que en su estado natural presentan alta multicolinealidad, el modelo estadístico necesita garantizar que cada nuevo eje capture una faceta del comportamiento comercial completamente distinta a las demás.

$$W^T W = I$$

Donde:

W^T = Representa la matriz transpuesta de los vectores propios (eigenvectores).

W = Representa la matriz original de los vectores propios.

I = Representa la Matriz Identidad (unos en la diagonal principal y ceros en el resto).

La ortogonalidad asegura que métricas distintas (ej. volumen vs intermitencia) no se contaminen mutuamente. Esta independencia matemática asegura que el algoritmo agrupe a los consumidores evaluando dimensiones puras, incrementando la eficacia de las decisiones comerciales.

10.6.1 Reducción de Dimensionalidad (Varianza Explicada)

La reducción de dimensionalidad es la finalidad ulterior del ACP, logrando sintetizar la información contenida en bases de datos masivas sin perder su riqueza analítica. La proporción de varianza explicada (PVE) es la métrica que cuantifica exactamente qué porcentaje de la dispersión total del sistema es capturada por un componente específico.

$$PVE = \lambda_i / \sum (j = 1 \text{ hasta } p) \lambda_j$$

Donde:

PVE = Representa la Proporción de Varianza Explicada por un componente individual.

λ_i = Representa el Eigenvalor o varianza capturada específicamente por el componente i .

$\sum \lambda_j$ = Representa la sumatoria de todos los Eigenvalores extraídos (varianza total).

Al emplear esta ecuación a la base de datos transaccional, se diseñarán los datos originales en un nuevo espacio reducido. Frenar las dimensiones que explican la gran mayoría de la varianza logrará que el algoritmo K-means procese la información con más precisión, eliminando el ruido de fondo.

10.7 Criterio de Kaiser

En la aplicación práctica de ACP, el Criterio de Kaiser es una regla matemática estandarizada diseñada para poder determinar el punto de corte exacto para la contención de componentes. Establece que un componente sintético solo justifica su existencia si es capaz de explicar mayor cantidad de información que una variable original individual estandarizada.

Retener si $\lambda_i > 1$ (aplicando $\max(n_comp, 2)$)

Donde:

λ_i = Representa el Eigenvalor asociado al componente principal evaluado.

1 = Representa la varianza unitaria teórica de una variable original estandarizada.

$\max$ = Representa la función que asegura retener un mínimo de 2 componentes para visualización.

En el código de Python ejecutado para esta tesis, la instrucción complementa el Criterio de Kaiser garantizando la retención de al menos dos ejes. Esto resulta de inmensa utilidad gerencial, ya que permite proyectar y visualizar los clústeres de clientes en gráficos de dispersión bidimensionales.

10.8 Agrupamiento K-Means

10.8.1 Método del Codo (Elbow Method)

Uno de los dilemas teóricos y desafíos prácticos más complejos al momento de diseñar e implementar metodologías de aprendizaje no supervisado, como lo es la segmentación algorítmica, radica en dictaminar la cantidad óptima de clústeres (grupos) sin poseer un conjunto de etiquetas de salida previamente establecidas. A diferencia del aprendizaje supervisado, aquí la máquina no sabe cuántos grupos de inventario existen en la realidad. Seleccionar una cantidad aleatoria de grupos resulta en un análisis defectuoso que impacta negativamente las políticas de compra.

Para resolver esta incertidumbre geométrica, el método heurístico del codo provee una solución analítica ampliamente avalada en la comunidad científica. Su propósito es evaluar matemáticamente la relación de compromiso (trade-off) directo entre la complejidad estructural del modelo estadístico (el número de grupos a manejar) y la precisión general del ajuste de los datos a dicho modelo (la reducción del error interno).

Este procedimiento no posee una única fórmula estática, sino que se fundamenta en la evaluación gráfica de la tasa de cambio y la derivada matemática del error, expresada conceptualmente así:

$$\textit{Tasa de Mejora Marginal} = \Delta J / \Delta k$$

El análisis del método se basa en el comportamiento de tres componentes teóricos principales.

Dónde

J : define la inercia total intraclúster del sistema, que es esencialmente la suma matemática de los errores o distancias cuadráticas de todos los productos hacia el centro de su respectivo grupo.

k : representa el número propuesto de clústeres a modelar (por ejemplo, particionar el inventario en 2, 3, 4 o más grupos). La expresión de la fracción.

$\Delta J / \Delta k$: expresa matemáticamente la derivada discreta o la tasa de disminución marginal de la inercia.

J : cada vez que el analista decide incrementar secuencialmente la cantidad de particiones (' k ').

El método requiere que el científico de datos itere el algoritmo repetidas veces, graficando el valor del error ' J ' en el eje de las ordenadas (Y) frente al número ascendente de clústeres ' k ' en el eje de las abscisas (X), buscando el cambio en el patrón de la curva.

Es un hecho estadístico indiscutible que, conforme el parámetro de particiones ' k ' incrementa en la gráfica, la inercia del sistema ' J ' disminuirá de manera ininterrumpida, ya que, al haber más centros, los puntos estarán más cerca de ellos. Sin embargo, al observar la relación ' $\Delta J / \Delta k$ ' existe un punto de inflexión estructural crítico en la gráfica donde la tasa de mejora marginal se desploma radicalmente, formando un ángulo visual que asemeja un 'codo' humano. Este codo indica que el modelo ha capturado la estructura principal de la base de datos de inventario. Traspasar este límite geométrico e incorporar particiones adicionales con el fin de disminuir el error conduce irremediablemente a un sobreajuste del modelo (overfitting). Un modelo sobre ajustado fragmentará artificialmente productos logísticos que en la realidad deberían tratarse con la misma política comercial. Por tanto, identificar este componente garantiza que el agrupamiento final del inventario no sea ni carente de profundidad analítica ni un ejercicio de complejidad innecesaria.

10.9 Función Objetivo K-Means (Inercia)

La alta eficacia predictiva, escalabilidad y robustez del algoritmo K-means como herramienta de segmentación se fundamentan enteramente en su avanzado proceso de optimización iterativa basado en el cálculo de distancias euclidianas. El núcleo del procedimiento radica en la aplicación del Algoritmo de Lloyd, una heurística diseñada para resolver el arduo problema matemático de clasificación multidimensional de datos.

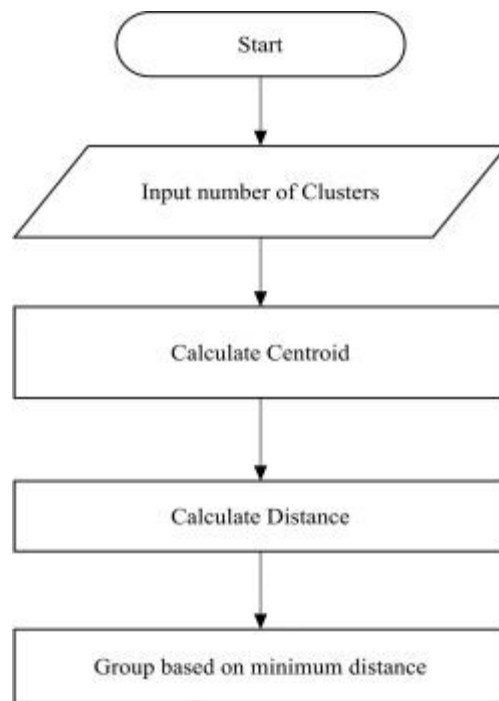


Ilustración 4 Diagrama de flujo del algoritmo K-means. Obtenido de "Prediction of leukemia by classification and clustering techniques"

A diferencia de las reglas empíricas de los gerentes, el algoritmo efectúa la segmentación mediante la minimización sistemática y constante de una Función Objetivo, que en el léxico formal de la inteligencia artificial y el aprendizaje automático es comúnmente denominada 'Inercia'. El objetivo final de la Inercia es asegurar que las observaciones (los productos del inventario) agrupadas en un mismo segmento posean características operativas, financieras y de rotación virtualmente idénticas, compactando el grupo al máximo y maximizando su densidad estadística en

el plano.

Dicha optimización se logra resolviendo un cálculo complejo que mide la cercanía espacial de los elementos, expresado formalmente a través de la siguiente fórmula matemática:

$$J = \sum_{i=1}^k \sum_{x \in C_i} ||x - \mu_i||^2$$

El desglose de esta función de costo revela el motor analítico del sistema.

Dónde:

J: Inercia total a minimizar, que actúa como el indicador del error de la segmentación.

K: señala la cantidad total de particiones o clústeres definidos previamente mediante el Método del Codo.

C_i: denota el subconjunto específico de variables o productos logísticos que el algoritmo ha asignado al clúster

i: durante la iteración actual. La letra minúscula

x: representa el vector matemático de las características estandarizadas del producto individual evaluado (conteniendo su demanda, costo, intermitencia, etc.). El término central

μ_i: corresponde a la coordenada del centroide matemático, que no es otra cosa que la media euclidiana de todos los puntos asignados al clúster

i: operador de la norma doble elevada al cuadrado,

$||x - \mu_i||^2$: se encarga de calcular y penalizar exponencialmente la distancia física entre el producto individual

x: centroide de su grupo

μ_i: sumando estas penalizaciones en un ciclo de optimización continua.

En el entorno operativo de la logística de almacenes y la gestión robusta de la cadena de suministro, alcanzar la convergencia algorítmica de esta extensa ecuación garantiza la obtención

final de agrupaciones sumamente compactas. En estas agrupaciones, las mercancías distribuirán perfiles comerciales de rotación completamente afines, asegurando congruencia en el análisis. El sistema reiterará el recálculo de esta fórmula cientos o miles de veces, reasignando productos de un grupo a otro hasta encontrar la ubicación adecuada de los centroides. Cuando el valor de la Inercia (J) logra estabilizarse en su mínima expresión matemática sin registrar variaciones en iteraciones subsiguientes, el sistema asume que ha encontrado la solución óptima. En este punto, el modelo computacional otorga una matriz categorizada definitiva que servirá a la directiva de la empresa como base científica indiscutible para definir estrategias corporativas, presupuestos de compras y políticas de abastecimiento, superando con creces los vetustos esquemas empíricos propensos al error humano.

K-Means es uno de los algoritmos de "clustering" más populares. K-significa almacena centroides k que utiliza para definir clústeres. Un punto se considera en un grupo particular si está más cerca del centroide de ese cúmulo que de cualquier otro centroide.

K-Means encuentra los mejores centroides alternando entre (1) asignar puntos de datos a los conglomerados basándose en los centroides actuales (2) elegir centroides (puntos que son el centro de un clúster) basándose en la asignación actual de puntos de datos a los clústeres.

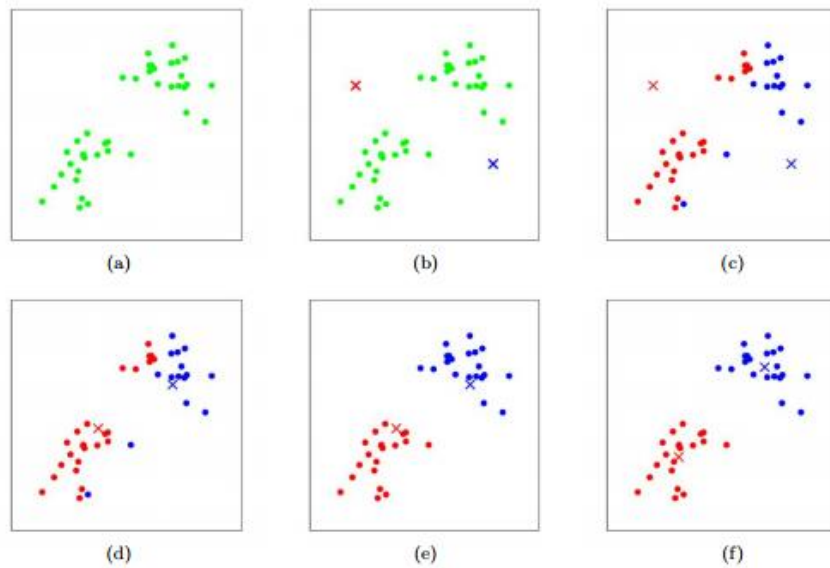


Ilustración 5 Ilustración del proceso iterativo de asignación y actualización de centroides en K-Means. Adaptado de "K-means" por C. Piech, Stanford University CS221 Handouts.

Algoritmo de K-medias. Los ejemplos de entrenamiento se muestran como puntos, y Los centroides de agrupación se muestran como cruces. (a) Conjunto de datos original. (b) Centroides iniciales aleatorios del grupo. (c-f) Ilustración de ejecutar dos iteraciones de k-medias. En cada uno Asignamos cada ejemplo de entrenamiento al centroide de clúster más cercano (mostrado al “pintar” los ejemplos de entrenamiento del mismo color que el grupo al que se le asigna el centroide); Luego movemos cada centroide de grupo a la media de los puntos asignados.

El algoritmo opera de la siguiente manera:

Inicialización: Elige k centroides iniciales de forma aleatoria (o por medio de un método más refinado como k-means)

Asignación: Cada punto de datos se asignará al centroide más cercano (esto se basa en la distancia euclidiana u otra métrica de distancia).

Actualización: Los centroides se actualizan, convirtiéndose en la media (centroide) de todos los puntos de datos asignados a ese clúster.

Convergencia: El segundo y tercer paso se repite hasta que la atribución de los puntos a los

cluster no cambia, o hasta alcanzar un máximo número de iteraciones.

Se entrena un modelo K-Means para cada k utilizando los scores de PCA del conjunto de entrenamiento (scores_train). N_init=20 significa que el algoritmo se ejecutará 20 veces con diferentes inicializaciones de centroides y elegirá la mejor configuración (la que tenga la menor inercia).

Se calcula el Coeficiente de Silueta, que viene a ser una métrica para evaluar la cohesión y división de los clusters. Un valor cercano a 1 significa que el objeto está bien agrupado; cercano a 0 significa solapamiento, y valores negativos significan que el objeto podría haber sido asignado a un clúster incorrecto. Se deberá maximizar esta métrica.

K ELEGIDO (4): Se escoge el número final de clústeres basándose en el análisis de métricas anteriores (por lo general se combina el método de codo y el coeficiente de silueta).

K means final: El modelo K means final que se ajusta con el K eleg

Labels_train, labels_test: Contienen el ID del clúster asignado a cada producto en los conjuntos de entrenamiento y prueba, respectivamente. Para la prueba, se usan los centroides aprendidos en el entrenamiento para pronosticar a qué clúster forma parte cada punto (asignación por centroide más cercano).

Las gráficas son un elemento fundamental para la selección de K:

Método del Codo (Inertia – izquierda): Esta muestra cómo la inercia disminuye a medida que aumenta K. Se busca un punto de “codo” donde la tasa de disminución de la inercia se vuelve marginal. Esto sugiere un k óptimo donde añadir más clústeres no aporta mucha más información.

Coeficiente de Silueta (derecha): Muestra el valor del coeficiente de silueta para cada K. Lo que se busca es el k que maximiza esta puntuación, lo que indica clústeres bien definidos y separados.

Para la **Inercia**, un “codo” podría observarse alrededor de $k=4$ o $k=5$. Para el Coeficiente de Silueta, el valor más alto está en $k=2$, pero luego hay otro pico alrededor de $k=4$. La elección de $K_ELEGIDO = 4$ es un balance coherente entre la inercia y la silueta, buscando obtener un número de clústeres que tenga sentido y sea interpretable.

10.10 Métricas de Evaluación de Clustering

Se usa coeficientes de silueta calculados para datos de entrenamiento y prueba, respectivamente. Un mayor valor muestra mejor calidad del clustering, con clusters más densos y separados de mejor forma. Se busca que los valores de entrenamiento y prueba sean parecidos, indicando que el modelo generaliza bien.

Las puntuaciones de Davies-Bouldin para los datos de entrenamiento y prueba miden la relación entre la dispersión dentro de los clústeres y la separación entre ellos. Un valor menor indica un mejor clustering, donde los clústeres son más compactos y están más separados entre sí. Al igual que con la silueta, se busca que los valores de entrenamiento y prueba sean consistentes.

La evaluación de estas métricas en los conjuntos de entrenamiento y prueba nos permitirá verificar la estabilidad del modelo y su capacidad para generalizar a datos no vistos.

10.10.1 Perfilado de Clústeres

Una vez que los productos han sido asignados a clústeres, el siguiente paso es comprender las características de cada grupo. El perfilado de clústeres se basa en calcular las estadísticas descriptivas (como la media o la mediana) de las variables originales para cada clúster.

Se agrupa el DataFrame resultado por el clúster asignado y se calculará la media de cada una de las 8 variables originales. Esto crea el perfil promedio para cada clúster.

Se añade una columna al perfil para contar cuántos productos hay en cada clúster, lo que da una idea del tamaño relativo de cada grupo.

Este perfil nos permite interpretar qué significa cada clúster en términos de las características originales de los productos, como su demanda, costo, rotación, etc.

El Data Frame perfil muestra las estadísticas descriptivas (la media en este caso) de las variables originales para cada uno de los clústeres identificados. También se incluye la columna n_productos que indica la cantidad de productos asignados a cada clúster. Esto permite caracterizar cada grupo de productos en función de sus atributos promedio.

11 METODOLOGÍA

11.1 ESTADÍSTICO DESCRIPTIVO

El objetivo del análisis descriptivo es resumir y caracterizar el comportamiento de las variables de la base de datos tanto categóricas como numéricas, mediante tablas de frecuencia y representaciones gráficas, sin realizar inferencia. La base de datos contiene 26.958 registros y 20 columnas.

Variables analizadas. El estudio se centra en cuatro variables categóricas y una numérica:

Tabla 2: Tipos de variables. Fuente: Elaboración propia.

Variable	Tipo	Descripción
Artículo	Categórica	Producto individual registrado en el kardex
Línea	Categórica	Línea o familia comercial a la que pertenece el producto
Documento	Categórica	Tipo de documento que respalda el movimiento
Persona	Categórica	Cliente o comprador asociado al movimiento
Precio	Numérica	Precio unitario del producto

Herramientas utilizadas

El análisis se desarrolló en Python (entorno Google Colab) empleando las siguientes librerías:

Pandas: manipulación y estructuración de los datos en DataFrames.

Numpy: manejo de valores nulos y operaciones numéricas (redondeo, generación de intervalos).

Matplotlib y Seaborn: construcción de los gráficos (barras, pastel e histograma).

11.2 Variables categóricas

Para obtener una representación gráfica se aplicaron los pasos estadísticos correspondientes para la obtención de tablas de frecuencia y adicionalmente las herramientas de creación de graficas:

Frecuencia absoluta: Haciendo uso de `value_counts()`, contamos las repeticiones de cada observación en su variable

Frecuencia relativa (%): Con `value_counts(normalize=True)*100`. Llevamos dicho conteo a una proporción de cada observación sobre el total.

Frecuencia acumulada: Utilizando (`cumsum()`) realizamos una suma progresiva de las frecuencias absolutas

Frecuencia relativa acumulada (%): suma progresiva de las frecuencias relativas.

Consolidación en una única tabla y representación gráfica.

Al presentarse el caso de variables con un alto conteo de categorías, siendo el caso de las variables de *Artículo* y *Persona*, se aplicó adicionalmente un agrupamiento Top 10 + «Otros», que concentra las diez categorías con mayores observaciones y el resto en una sola categoría. Esto mejora la legibilidad del gráfico sin perder la información de las categorías dominantes.

11.3 Variable numérica

Para la variable Precio se elaboró una tabla de frecuencias de datos agrupados. Primero se eliminaron los valores nulos. El número de clases se determinó con la regla de Sturges:

$$k = 1 + 3,322 \cdot \log_{10}(n)$$

A partir de k se generaron intervalos de igual amplitud entre el valor mínimo y máximo del precio, se asignó cada observación a su clase (`pd.cut`) y se calcularon las frecuencias absoluta, relativa y acumulada. Complementariamente se usó `df.describe()` para obtener las medidas de resumen (media, desviación estándar, mínimo, cuartiles y máximo) y un histograma con curva de densidad (KDE) para visualizar la forma de la distribución.

12 ANÁLISIS DE RESULTADOS

12.1 Caracterización descriptiva de la base de datos

Antes de construir el modelo se caracteriza por el comportamiento de las variables registradas en nuestra base de datos (kárdex), con el propósito de conocer la estructura de los datos de partida.

12.2 Variable Artículo (Producto)

La variable Artículo presenta una gran cantidad de categorías (productos distintos). La tabla siguiente muestra los diez productos más frecuentes y la categoría «Otros», que agrupa el resto.

Tabla 3: Frecuencias de los 10 productos más vendidos. Fuente: Elaboración propia.

Producto (Top 10)	Frec. Abs.	Frec. Rel. (%)	Frec. Rel. Acum. (%)
PEGANTE ESPAÑOL STANDARS 20K	3.173	11,77	11,77
PEGANTE ESPAÑOL PREMIUM 20K	2.430	9,01	20,78
PEGANTE ESPAÑOL PLUS 20K	1.559	5,78	26,57
PORCELANA GRIS CLARO 2K	1.508	5,59	32,16
PORCELANA ALMENDRA 2K	1.312	4,87	37,03
PORCELANA BLANCO 2K	1.290	4,79	41,81
PEGANTE ESPAÑOL PREMIUM 25K	803	2,98	44,79
PORCELANA GRIS CLARO 1K	764	2,83	47,63
PORCELANA GRIS OSCURO 2KG	654	2,43	50,05
PORCELANA ALMENDRA 1K	643	2,39	52,44
Otros	12.822	47,44	—

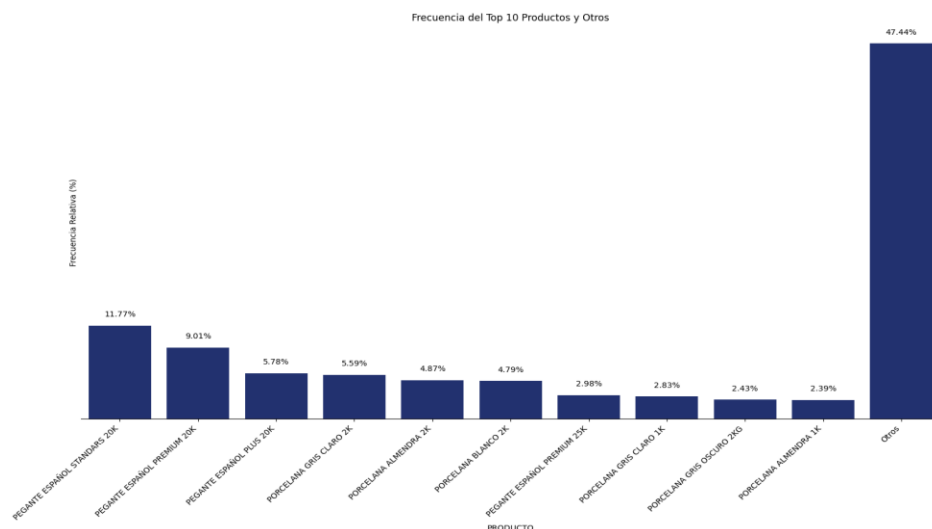


Ilustración 6 Frecuencia relativa del Top 10 de productos y «Otros».

Como se puede apreciar, el 52,40% de los productos vendidos se encuentran concentrados en el top 10, mientras que el 47,4 % restante se reparte entre un gran número de productos distintos. El producto con mayor venta es el PEGANTE ESPAÑOL STANDARS 20K, con un 11,77%. Se observa además que los pegantes y las porcelanas dominan claramente el catálogo de mayor rotación, lo que sugiere que la operación se concentra en un conjunto reducido de productos de alta demanda.

12.3 Variable Línea

La variable Línea agrupa los productos en siete categorías comerciales. Al tener pocas categorías, se representa mediante un gráfico de pastel.

Tabla 4: Frecuencias de las líneas de productos. Fuentes: Elaboración propia.

Línea	Frec. Abs.	Frec. Rel. (%)	Frec. Rel. Acum. (%)
EMPORE	13.086	48,54	48,54
PEGANTE	8.793	32,62	81,16
MATERIALES	3.558	13,20	94,36
POLVO MINERAL	781	2,90	97,25
EMPASTE	426	1,58	98,84
BLANQUEADOR	197	0,73	99,57
SERVICIOS	117	0,43	100,00

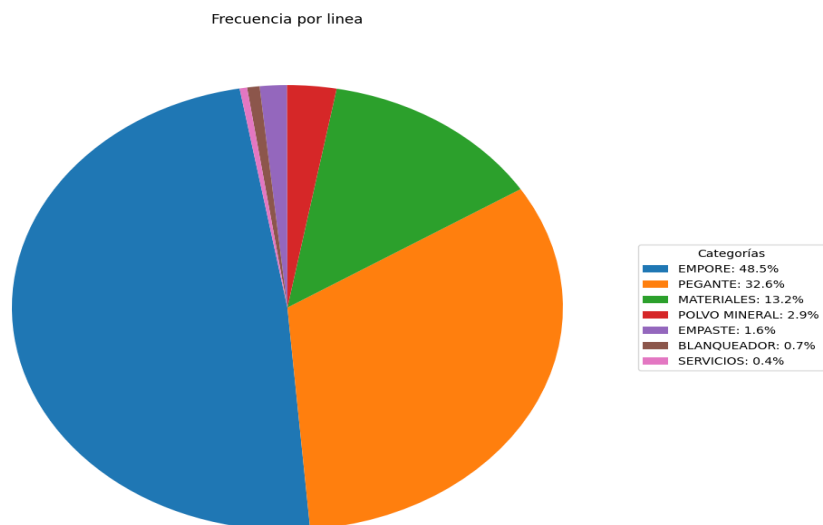


Ilustración 7 Distribución de frecuencia por Línea.

En relación con los productos que presentan mayor movimiento, las líneas EMPORE (48,54 %) y PEGANTE (32,62 %) son las que concentran la mayor parte de las operaciones, sumando entre ambas el 81,16 % del total. Al incluir MATERIALES, con el 13,20 %, estas tres líneas representan el 94,36 % de la actividad. Por otro lado, POLVO MINERAL, EMPASTE, BLANQUEADOR y SERVICIOS tienen una participación mucho menor, ya que ninguna supera el 3 %. En general, los resultados muestran que la mayor parte de la actividad del negocio está relacionada con los productos de emporado y porcelana, junto con los pegantes.

12.4 Variable Documento

La variable Documento describe el tipo de comprobante que respalda cada movimiento. Presenta solo cuatro categorías.

Tabla 5: Frecuencias por tipos de documentos. Fuentes: Elaboración propia.

Documento	Frec. Abs.	Frec. Rel. (%)	Frec. Rel. Acum. (%)
FACTURA	17.970	70,71	70,71
COMPROBANTE DE VENTA	5.089	20,03	90,74
COMPRA	2.229	8,77	99,51
NOTA DE CREDITO	124	0,49	100,00

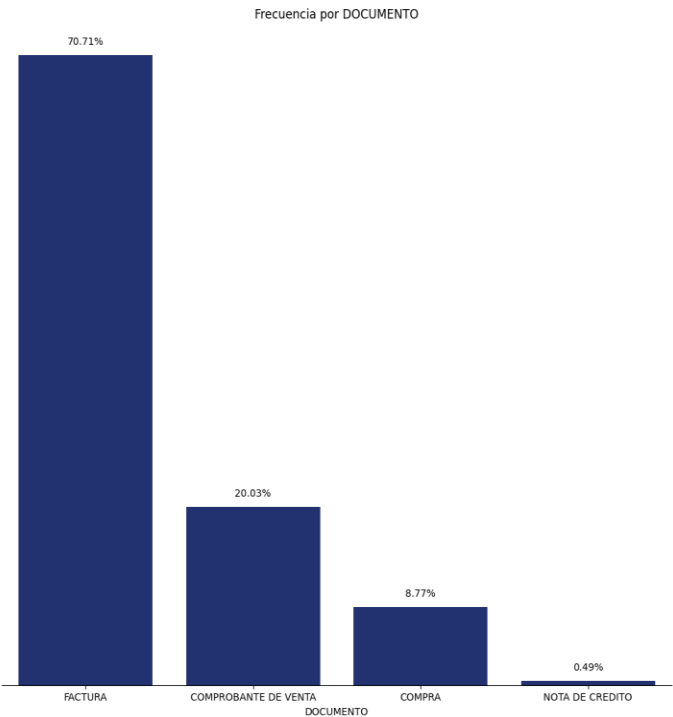


Ilustración 8 Frecuencia relativa por tipo de Documento.

Al ser un negocio dedicado principalmente a la comercialización de productos, es normal que los documentos que más se manejen sean las FACTURAS, que representan el 70,71 % de los registros, seguidas de los COMPROBANTES DE VENTA, con el 20,03 %. Entre ambos documentos concentran más del 90 % de las operaciones, lo que refleja que la actividad está enfocada principalmente en las ventas y en el movimiento constante del inventario. Por otro lado, las COMPRAS representan el 8,77 % de los movimientos, mientras que las NOTAS DE CRÉDITO tienen

una participación bastante baja, con apenas el 0,49 %. Esto puede considerarse un aspecto positivo, ya que refleja un nivel reducido de devoluciones, anulaciones o ajustes en las operaciones.

12.5 Variable Persona (Compradores)

La variable Persona nos sirve para identificar a los compradores; se decidió realizar una agrupación de los 10 máximos y del resto, debido a su alta cantidad de observaciones.

Tabla 6: Frecuencias de compradores. Fuente: Elaboración propia.

Comprador (Top 10)	Frec. Abs.	Frec. Rel. (%)	Frec. Rel. Acum. (%)
MACIAS CABASCANGO GAUDENCIO	1.557	6,13	6,13
REMACHE GUAPIZACA MANUEL J. / COMERCIAL ANITA	826	3,25	9,38
ULLAURI MACIAS ANA CECIBEL / JORGE NAJERA	755	2,97	12,35
VELOZ VILLAMAR JOSE MANUEL	688	2,71	15,06
BAZURTO QUIROZ ELIANA ANTONIA	543	2,14	17,19
ACOSTA ROCAFUERTE KAREN E. / DECOCERAMI...	513	2,02	19,21
ALCIVAR HOLGUIN SAMMY NARCISA	507	2,00	21,21
CULQUI VISCAINO MELANIE PAOLA	494	1,94	23,15
CABRERA GODOY GABRIELA ISABEL	493	1,94	25,09
COMERCIAL LINDA / RIVERA CANTOS MARTHA J.	460	1,81	26,90
Otros	18.576	73,10	—

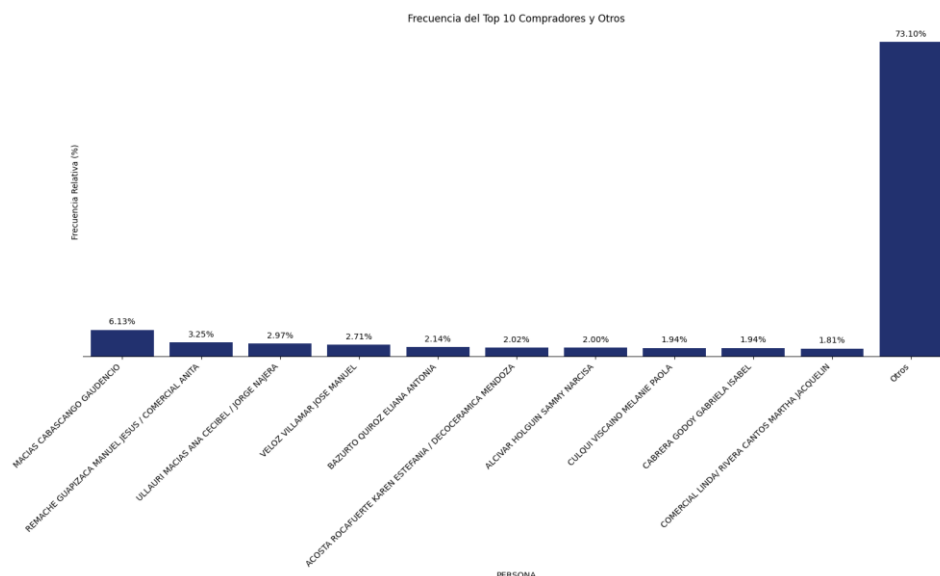


Ilustración 9 Frecuencia relativa del Top 10 de compradores y «Otros».

El cliente que registra más movimientos es MACIAS CABASCANGO GAUDENCIO, aunque su participación es de apenas el 6,13 % del total. Si se consideran los diez clientes con mayor cantidad de movimientos, estos representan el 26,9 %, mientras que el 73,1 % restante se distribuye entre una gran cantidad de compradores. Esto demuestra que el negocio tiene una cartera de clientes bastante amplia y que no depende de un grupo reducido para generar sus ventas. Aun así, sería conveniente buscar estrategias que permitan mantener a estos clientes y motivarlos a realizar sus compras con mayor frecuencia.

12.6 Variable numérica Precio

Aplicando la regla de Sturges sobre los 25.412 precios válidos se obtuvieron 16 clases de igual amplitud. La tabla resume la distribución por intervalos de precio.

Tabla 7: Frecuencias por intervalo de precios. Fuentes: Elaboración propia.

Intervalo de Precio	Frec. Abs.	Frec. Rel. (%)	Frec. Acum. (%)
[0,00 – 6,25)	23.941	94,21	94,21
[6,25 – 12,50)	676	2,66	96,87
[12,50 – 18,75)	296	1,16	98,03
[18,75 – 25,00)	323	1,27	99,30
[25,00 – 31,25)	42	0,17	99,47
[31,25 – 37,50)	45	0,18	99,65
[37,50 – 43,75)	12	0,05	99,70
[43,75 – 50,00)	24	0,09	99,79
[50,00 – 62,50)	0	0,00	99,79
[62,50 – 68,75)	37	0,15	99,94
[68,75 – 100,00)	12	0,04	99,98



Ilustración 10 Frecuencia relativa por intervalos de Precio.

El catálogo maneja principalmente productos de precios bajos, ya que el 94,21 % tiene un valor inferior a 6,25 unidades monetarias. Además, cerca del 99,3 % de los productos se encuentra por debajo de 25, por lo que este es el rango de precios que se maneja con mayor frecuencia en el negocio.

A partir de estos valores, la cantidad de productos disminuye considerablemente y los precios superiores a 50 son poco comunes. En general, se puede decir que el negocio trabaja principalmente con productos de bajo costo unitario, concentrando la mayor parte de su oferta en precios menores a 25.

12.6.1 Medidas de resumen

Las medidas descriptivas de la variable Precio (df.describe()) confirman lo observado en la tabla de frecuencias:

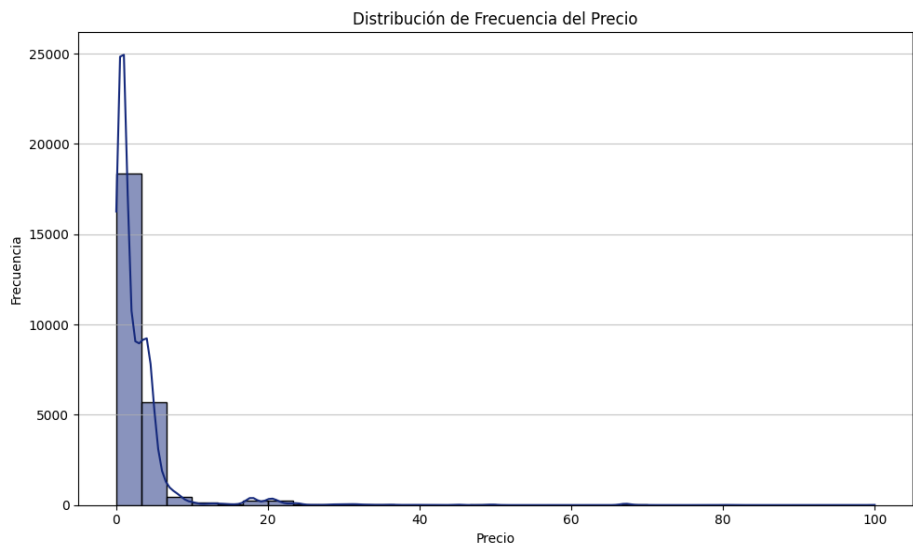


Ilustración 11 Histograma y densidad (KDE) de la distribución del Precio.

	codigo	Categoria	Familia	Unnamed: 9	#	+	-	Saldo	costo Ultimo	costo Promedio	Precio	valor
count	28958.000000	0.0	0.0	0.0	2.541200e+04	3.875000e+03	24522.000000	2.612600e+04	26126.000000	26126.000000	25412.000000	2.695800e+04
mean	79.667887	NaN	NaN	NaN	8.068052e+04	8.181176e+04	174.961122	6.140404e+06	1.348857	1.166113	2.730935	9.013947e+06
std	68.921854	NaN	NaN	NaN	2.678762e+06	1.967413e+06	2839.254197	1.626156e+07	6.801226	6.513446	5.090242	2.421704e+07
min	0.148000	NaN	NaN	NaN	1.000000e+00	0.000000e+00	0.000000	-1.590900e+04	-66.666700	-66.666700	0.000000	-2.497560e+02
25%	16.000000	NaN	NaN	NaN	8.810000e+02	2.000000e+01	15.000000	-8.137500e+01	0.000000	0.000000	0.848200	7.545450e+02
50%	91.000000	NaN	NaN	NaN	2.157000e+03	8.000000e+01	30.000000	2.239400e+04	0.000000	0.000000	0.950000	1.426136e+04
75%	95.000000	NaN	NaN	NaN	9.481750e+03	2.500000e+02	60.000000	4.817685e+05	1.826800	1.600000	3.704800	8.661754e+05
max	346.000000	NaN	NaN	NaN	9.800345e+07	5.002793e+07	231654.000000	4.996153e+07	413.816000	413.816000	100.000000	7.540694e+07

Ilustración 12: Medidas descriptivas de las variables numéricas

Ilustración 13: Medidas descriptivas de las variables numéricas. Fuente: Elaboración propia.

Medida	Precio	valor
N (válidos)	25.412	26.958
Media	2,73	9.013.947
Desv. estándar	5,09	24.217.040
Mínimo	0,00	-249,76
Q1 (25 %)	0,85	754,55
Mediana (50 %)	0,95	14.261,36
Q3 (75 %)	3,70	866.175,40
Máximo	100,00	75.406.940

La media del precio es de 2,73, mientras que la mediana es de 0,95, por lo que existe una diferencia importante entre ambos valores. La desviación estándar es de 5,09, incluso mayor que el promedio. Esto se explica porque la mayoría de los productos tiene precios bajos: el 75 % tiene un valor de 3,70 o menos. Sin embargo, también existen algunos productos con precios mucho más altos, llegando hasta 100, lo que hace que el promedio aumente. Por esta razón, la mediana de 0,95 refleja de mejor manera el precio que normalmente se encuentra en los productos del negocio.

En cuanto al monto de los movimientos, la diferencia entre los valores es todavía más evidente. La mediana es de aproximadamente 14.261, mientras que el promedio llega a cerca de 9,01 millones y el valor máximo alcanza aproximadamente 75,4 millones. La desviación estándar también es bastante alta, con alrededor de 24,2 millones, lo que muestra que los montos registrados son muy variados. En la mayoría de los casos se realizan movimientos por valores relativamente pequeños, pero existen algunas operaciones de montos muy altos que hacen que el promedio se eleve considerablemente. Por otra parte, el valor mínimo de -249,76 podría estar relacionado con una nota de crédito, devolución o algún ajuste registrado en el sistema.

12.7 Construcción y ejecución del modelo

Para la construcción del modelo, se realizará con Google Colab, una plataforma en la nube que nos va a permitir ejecutar código Python. Para proceder, es fundamental contar con una base de

datos preparada, siendo recomendable trabajar con variables numéricas estandarizadas, ya que tanto K-means como el ACP son sensibles a la escala de las variables (Jolliffe & Cadima, 2016).

12.8 Paso 1.- Preparación del Entorno de Trabajo

Llamamos a las Librerías

Sección 1.1: Librerías de manipulación y visualización

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Las librerías a importar van a depender del algoritmo y la base de datos a utilizar. A continuación, se explicará detalladamente cada una de las librerías empleadas en este trabajo de investigación.

Pandas: Es una librería de código abierto utilizada para la manipulación y análisis de datos, diseñadas para facilitar el trabajo con datos relacionales o etiquetados de manera intuitiva. Permite realizar operaciones de lectura, limpieza, transformación el análisis de datos estructurados procedentes de diversas fuentes (Andersen, 2008).

NumPy: Es la librería fundamental para la computación numérica en Python. Proporciona un objeto de arreglo múltiple de alto rendimiento, junto con herramientas para trabajar con dichos arreglos, incluyendo funciones matemáticas, álgebra lineal y generación de números aleatorios (Harris, 2020).

Matplotlib: Es una librería de visualización que produce gráficos estáticos, animados e interactivos. Permite un control detallado sobre cada elemento visual (ejes, colores, leyendas, anotaciones), lo cual resulta fundamental para representar gráficamente la varianza explicada del ACP y los resultados del método del codo y la silueta (Hunter, 2007)

Seaborn: Construida sobre matplotlib, esta librería facilita la creación de gráficos estadísticos más elaborados con una sintaxis simplificada.

Scikit-learn (sklearn): Es una librería de aprendizaje automático que proporciona herramientas simples y eficientes para el análisis predictivo y no supervisado de datos, construida sobre NumPy, SciPy y matplotlib (Pedregosa, 2011). De esta librería se emplearon específicamente los siguientes módulos:

StandardScaler: estandariza las variables (media 0, desviación estándar 1), requisito indispensable antes de aplicar tanto ACP como K-means, ya que ambos algoritmos son sensibles a la escala de las variables.

12.8.1 Módulos de aprendizaje automático

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, silhouette_samples, davies_bouldin_score
```

Esta sección importa selectivamente los componentes de *Scikit-learn* requeridos por el modelo, en lugar de la librería completa, práctica que optimiza el consumo de memoria del entorno (Pedregosa et al., 2011). En este caso, la función *train_test_split* da paso a la partición del conjunto

de datos; *StandardScaler* ejecutando la estandarización de variables; PCA implementa el Análisis de Componentes Principales; K-Means corresponde al algoritmo de agrupamiento particional; y del módulo de métricas se incorporan el coeficiente de silueta (tanto en su versión promedio como por observación individual) y el índice de Davies-Bouldin, empleados para la validación interna de la segmentación.

12.8.1.1 Opciones de repetición de resultados y colores

```
np.random.seed(42)

MAROON, BLUE, GOLD, GREEN = '#8B1E3F', '#1B3A6B', '#C9A227', '#4C9A6A'

PALETTE = [MAROON, BLUE, GOLD, GREEN]
```

np.random.seed(42) fija la semilla del generador de números semialeatorios del entorno, garantizando que la partición de datos y la inicialización de centroides produzca resultados idénticos en cada ejecución del programa. Las tres líneas restantes definen los códigos de una paleta institucional de cuatro colores, almacenada en la lista *PALETTE*, que será utilizada más adelante para diferenciar visualmente los resultados obtenidos.

12.9 Paso 2: Carga y limpieza de los datos

Se agrega el archivo con los datos

12.9.1 Transferencia del archivo al entorno de trabajo de Google Colab

```
#Cargamos datos desde la Pc

from google.colab import files

uploaded = files.upload()
```

El módulo *files* de la librería *google.colab* permite subir archivos hacia el entorno de trabajo

alojado en la nube.

`files.upload()` despliega un control de selección de archivos en la interfaz del cuaderno y almacena en el diccionario `uploaded` los nombres y contenidos de los archivos transferidos.

12.9.2 Lectura y normalización de encabezados

```
sheet_name='Super_Kardex_General', header=2)

df = df.rename(columns={'2023-12-31 00:00:00.000': 'fecha',
                        'CEMENTO BLANCO HUASCARAN (50KG.)': 'producto'})

df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
```

La función **`pd.read_excel()`** incorpora al entorno la hoja denominada `Super_Kardex_General` del archivo de kárdex correspondiente al período 2023–2025. El parámetro `header=2` indica que los encabezados efectivos de la tabla se localizan en la tercera fila del archivo. Luego, el método `rename()` que corrige dos encabezados que, por efecto de dicho formato, habían sido interpretados como valores de datos en lugar de nombres de columna. Finalmente, `pd.to_datetime()` convierte la columna de fecha al tipo temporal nativo de pandas; el argumento `errors='coerce'` asigna el valor nulo `NaT` a los registros cuya fecha no admita conversión, evitando la interrupción de la ejecución (McKinney, 2010).

12.9.3 Depuración de datos duplicados

12.9.3.1 Identificación y eliminación de duplicados

```
cols_check = [c for c in df.columns if c not in ['Unnamed: 9']]

filas_antes = df.dropna(subset=['producto']).shape[0]

df = df.drop_duplicates(subset=cols_check).copy()

filas_despues = df.dropna(subset=['producto']).shape[0]
```

```
print(f"Filas con producto antes de deduplicar : {filas_antes}")  
print(f"Filas con producto después de deduplicar: {filas_despues}")
```

La primera instrucción construye el conjunto de columnas que servirá como criterio de comparación, sin contar a la columna *Unnamed: 9*.

Luego, se registra el número de filas con producto identificado antes de la depuración, se aplica *drop_duplicates()* para eliminar aquellas filas idénticas.

```
Filas con producto antes de duplicar: 26956  
Filas con producto después de duplicar: 17863
```

El resultado evidencia la eliminación de 9 093 registros duplicados, equivalentes al 33.7% del volumen original.

12.10 Paso 3: Construcción de las variables

12.10.1 Conversión de tipos y tabla maestra de productos

```
tx = df.dropna(subset=['producto','fecha']).copy()  
tx['mes'] = tx['fecha'].dt.to_period('M')  
tx['SALDO ANTERIOR'] = pd.to_numeric(tx['SALDO ANTERIOR'], errors='coerce')  
tx[28.6071] = pd.to_numeric(tx[28.6071], errors='coerce')  
tx['28.6071.1'] = pd.to_numeric(tx['28.6071.1'], errors='coerce')  
tx[1859.464] = pd.to_numeric(tx[1859.464], errors='coerce')  
  
master = df.dropna(subset=['producto']).groupby('producto').agg(  
    Articulo=('producto','first'), Linea=('MATERIALES','first')).reset_index()
```

Se construye el objeto tx como una copia del conjunto limpio, restringida a los registros que poseen simultáneamente producto y fecha válidos. `dt.to_period('M')` genera la variable de período mensual, indispensable para la agregación temporal posterior. Las cuatro instrucciones siguientes ayudan la conversión a tipo numérico de los campos de saldo, costo promedio, último costo y precio de venta, asignando valores nulos a las entradas no convertibles. Cabe señalar que tres de estos campos son referenciados mediante identificadores numéricos, circunstancia derivada de que el archivo de origen no consignó denominaciones textuales en dichas columnas. Finalmente, se construye la tabla maestra master, que consolida la denominación del artículo y su línea de producto.

12.10.2 Matriz de demanda mensual por producto

```
mensual = tx.groupby(['producto','mes'])[0.148].sum().reset_index()
all_months = pd.period_range(tx['mes'].min(), tx['mes'].max(), freq='M')
piv = mensual.pivot(index='producto', columns='mes', values=0.148)\
reindex(columns=all_months, fill_value=0).fillna(0)
```

Esta sección genera la estructura central del análisis. La primera instrucción agrega las cantidades despachadas por producto y mes mediante `groupby()` y `sum()`. La segunda construye la secuencia completa de períodos mensuales comprendidos entre la primera y la última transacción registrada.

La tercera organizará nuevamente la información en una matriz de dos entradas (productos en filas y meses en columnas) mediante `pivot()`, reorganizado sobre la secuencia total de meses y asignando el valor cero a los períodos donde no se refleje movimiento. Esta atribución resulta metodológicamente necesaria: los meses sin demanda constituyen información sustantiva sobre el comportamiento del producto, y su omisión distorsionaría el cálculo de la intermitencia.

12.10.3 Variables de demanda, variabilidad e intermitencia

```
demanda_total = piv.sum(axis=1)

media_mensual = piv.mean(axis=1)

std_mensual = piv.std(axis=1)

cv_demanda = (std_mensual / media_mensual.replace(0, np.nan)).fillna(0)

intermitencia = (piv == 0).mean(axis=1)

frecuencia = tx[tx[0.148]>0].groupby('producto').size().reindex(demanda_total.index, fill_value=0)
```

A partir de la matriz producto-mes se derivan cinco indicadores mediante operaciones vectorizadas sobre el eje horizontal (axis=1). La demanda total es la suma de los despachos durante todo el período. La media y la desviación estándar mensual sirven para calcular el coeficiente de variación como su cociente. Antes de esto, se eliminan los denominadores nulos y se reemplazan por valores faltantes para evitar divisiones indeterminadas. Se calcula la intermitencia como la proporción de meses con demanda nula usando la función mean() sobre una matriz booleana. Por último, la frecuencia de salidas cuenta el número de transacciones con cantidad positiva para cada producto. Estos indicadores constituyen las dimensiones clásicas empleadas en la clasificación multicriterio de inventarios (Silver et al., 2016).

12.10.4 Variables de saldo y rotación

```
saldo_prom = tx.groupby('producto')['SALDO ANTERIOR'].mean()

pct_saldo_neg = tx.groupby('producto').apply(

lambda g: (g['SALDO ANTERIOR']<0).mean(), include_groups=False)

rotacion = (demanda_total / saldo_prom.replace(0, np.nan)).fillna(0)
```

Se calcula el saldo promedio de inventario por producto y la proporción de registros con saldo negativo, indicador este último que permite detectar inconsistencias en el control físico de

existencias. La rotación de inventario se obtiene como el cociente entre la demanda total y el saldo promedio. Es importante mencionar que, según se documenta en la fase siguiente, el campo de saldo del kárdex no contenía registros válidos, circunstancia que derivó en la exclusión efectiva de ambas variables del modelo y que constituye una limitación declarada del presente estudio.

12.10.5 Costo unitario mediante cascada de validez

```
costo_prom = tx[tx[28.6071]>0].groupby('producto')[28.6071].mean()
costo_ult = tx[tx['28.6071.1']>0].groupby('producto')['28.6071.1'].mean()
precio = tx[tx[1859.464]>0].groupby('producto')[1859.464].mean()

costo_final = costo_prom.reindex(demanda_total.index).fillna(costo_ult).fillna(precio)
valor_consumido = demanda_total * costo_final
```

Este bloque implementa un procedimiento de imputación jerárquica denominado cascada de validez. Se calculan de forma independiente tres estimadores del valor unitario: el costo promedio, el último costo registrado y el precio de venta, considerando en cada caso únicamente los registros con valores positivos. Luego, el encadenamiento de métodos `fillna()` constituir la prelación: se privilegia el costo promedio y, ante su ausencia, se recurre sucesivamente al último costo y al precio. Esta estrategia reduce la pérdida de observaciones por datos ausentes en un único campo. El valor consumido resulta del producto entre la demanda total y el costo unitario, y constituye la variable central de los esquemas de clasificación ABC (Lolli, 2013).

12.10.6 Consolidación de la matriz de características

```
feat = pd.DataFrame({
    'demanda_total': demanda_total, 'frecuencia_salidas': frecuencia,
    'cv_demanda': cv_demanda, 'intermitencia': intermitencia,
```

```

'costo_unit_prom': costo_final, 'valor_consumido': valor_consumido,
'saldo_promedio': saldo_prom, 'rotacion_inventario': rotacion,
'pct_costo_valido': pct_costo_valido, 'pct_saldo_negativo': pct_saldo_neg,
}).reset_index().merge(master, on='producto', how='left')

feat = feat.replace([np.inf,-np.inf],
np.nan).dropna(subset=['costo_unit_prom']).reset_index(drop=True)

```

Las series calculadas se consolidan en un único DataFrame denominado feat, en el cual cada fila representa un producto y cada columna una característica de su comportamiento. La operación merge() incorpora la información descriptiva de la tabla maestra mediante una unión por la izquierda, preservando la totalidad de los productos analizados. La instrucción final sustituye los valores infinitos —originados en divisiones por cero— por valores faltantes y descarta los productos que carecen de costo unitario válido, por cuanto dicha variable resulta indispensable para el cálculo del valor consumido.

12.11 Paso 4: Transformación, Partición y Estandarización

12.11.1 Definición de variables y transformaciones

```

vars8 = ['demanda_total','frecuencia_salidas','cv_demanda','intermitencia',
'costo_unit_prom','valor_consumido','saldo_promedio','rotacion_inventario']

log_vars =
['demanda_total','frecuencia_salidas','costo_unit_prom','valor_consumido','saldo_promedio']

X = feat[vars8].copy()

for v in log_vars:

```



```
X[v] = np.log1p(X[v].clip(lower=0))
```

```
X['rotacion_inventario'] = np.arcsinh(X['rotacion_inventario']) # puede ser negativo
```

Se establece el vector de las ocho variables que incorporar el modelo y el subconjunto sobre el cual se aplicará la transformación logarítmica. El ciclo participativo aplica la función `log1p` equivalente a $\ln(1+x)$ a cinco variables que presentan distribuciones marcadamente asimétricas, producto de la coexistencia de productos de alta y baja rotación en el mismo portafolio; el método `clip(lower=0)` previene la aparición de argumentos negativos. La variable de rotación recibe un tratamiento diferenciado mediante la función arcoseno hiperbólico inverso, dado que admite valores negativos y, en consecuencia, no resulta compatible con la transformación logarítmica convencional.

Fuente especificada no válida..

12.11.2 Partición en subconjuntos de entrenamiento y prueba

```
train_idx, test_idx = train_test_split(feat.index, test_size=0.2, random_state=42)
```

```
X_train, X_test = X.loc[train_idx], X.loc[test_idx]
```

```
feat_train, feat_test = feat.loc[train_idx].copy(), feat.loc[test_idx].copy()
```

La función `train_test_split()` divide aleatoriamente los índices del conjunto de productos, destinando el 80 % al entrenamiento del modelo y el 20 % restante a su validación. El parámetro `random_state=42` asegura que la partición sea idéntica en cada ejecución. Los índices resultantes se emplean para segmentar tanto la matriz transformada como el conjunto original de características, este último requerido para el perfilamiento posterior de los conglomerados en las unidades de medida originales.

12.11.3 Tratamiento de valores faltantes

```
cols_to_drop = []

for col in X_train.columns:
    if X_train[col].isnull().any():
        median_val = X_train[col].median()
        if pd.isna(median_val):
            cols_to_drop.append(col)
        else:
            X_train[col] = X_train[col].fillna(median_val)
            X_test[col] = X_test[col].fillna(median_val)
    if cols_to_drop:
        print(f"Columnas completamente NaN en X_train y serán eliminadas: {cols_to_drop}")
X_train = X_train.drop(columns=cols_to_drop)
X_test = X_test.drop(columns=cols_to_drop)
```

El ciclo recorre cada variable cerciorando la presencia de valores ausentes. Cuando estos existen, se calcula la mediana del conjunto de entrenamiento y se la aplica como valor de atribución en ambos subconjuntos; se trabaja con la mediana sobre la media por su robustez frente a valores extremos. Si la mediana resulta indefinida (situación que indica que toda la columna carece de datos válidos), la variable se registra para su eliminación. Es metodológicamente importante que el estadístico de imputación se calcule exclusivamente sobre el conjunto de entrenamiento, evitando la transferencia de información desde el conjunto de prueba.

```
Columnas completamente NaN en X_train y serán eliminadas: ['saldo_promedio']
```

La salida documenta la exclusión de la variable saldo_promedio por ausencia total de registros válidos en el campo correspondiente del kárdex. Esta circunstancia reduce a siete el número de variables efectivas del modelo y anula el poder discriminante de la rotación de inventario, hallazgo que se declara limitación del estudio.

12.11.4 Estandarización de variables

```
scaler = StandardScaler().fit(X_train)      # fit SOLO en train  
Xz_train, Xz_test = scaler.transform(X_train), scaler.transform(X_test)
```

El objeto StandardScaler se ajusta únicamente sobre el conjunto de entrenamiento mediante el método fit(), calculando la media y la desviación estándar de cada variable; posteriormente, el método transform() aplica la estandarización a ambos subconjuntos. Este orden de operaciones previene la fuga de información hacia el conjunto de prueba. La estandarización constituye un requisito inevitable por cuanto tanto el Análisis de Componentes Principales como el algoritmo K-means operan sobre distancias euclidianas, sensibles a las unidades de medida de las variables; su omisión permitiría que variables expresadas en unidades monetarias dominaran el cálculo frente a variables expresadas como proporciones (Wongoutong, 2024).

12.12 Paso 5: Reducción de Dimensionalidad

12.12.1 Ajuste del Análisis de Componentes Principales

```
#ACP  
pca = PCA(random_state=42).fit(Xz_train)  
evr, eigvals = pca.explained_variance_ratio_, pca.explained_variance_  
  
n_comp = max(int((eigvals>1).sum()), 2) # criterio de Kaiser
```

El objeto PCA se ajusta sobre la matriz estandarizada de entrenamiento, procedimiento mediante el cual se calculan los autovalores y autovectores de la matriz de covarianzas. Los atributos `explained_variance_ratio_` y `explained_variance_` almacenan, respectivamente, la proporción de varianza explicada y los autovalores asociados a cada componente. La determinación del número de componentes a retener aplica el criterio de Kaiser, conservando aquellos cuyo autovalor supera la unidad; el uso de la función `max()` con el valor dos establece un mínimo que garantiza la posibilidad de representación gráfica bidimensional (Jolliffe, 2002).

12.12.2 Matriz de cargas factoriales y proyección de los datos

```
loadings = pd.DataFrame(pca.components_[0:n_comp].T, index=X_train.columns,
                        columns=[f'CP{i+1}' for i in range(n_comp)])

scores_train_full = pca.transform(Xz_train)
scores_test_full = pca.transform(Xz_test)
scores_train, scores_test = scores_train_full[:,0:n_comp], scores_test_full[:,0:n_comp]
```

La matriz de cargas se construye transponiendo los autovectores correspondientes a los componentes retenidos e indexándolos con los nombres de las variables originales, lo que permite identificar la contribución de cada variable a cada componente. Seguidamente, el método `transform()` proyecta ambos subconjuntos sobre el nuevo sistema de coordenadas, y la operación de segmentación conserva únicamente las columnas correspondientes a los componentes seleccionados. Los objetos `scores_train` y `scores_test` constituyen la entrada efectiva del algoritmo de agrupamiento.

12.12.3 Representación gráfica de la varianza explicada

```
fig, ax = plt.subplots(1, 2, figsize=(14, 5))
ax[0].plot(range(1, len(evr) + 1), evr, marker='o', linestyle='--', color=MAROON)
```

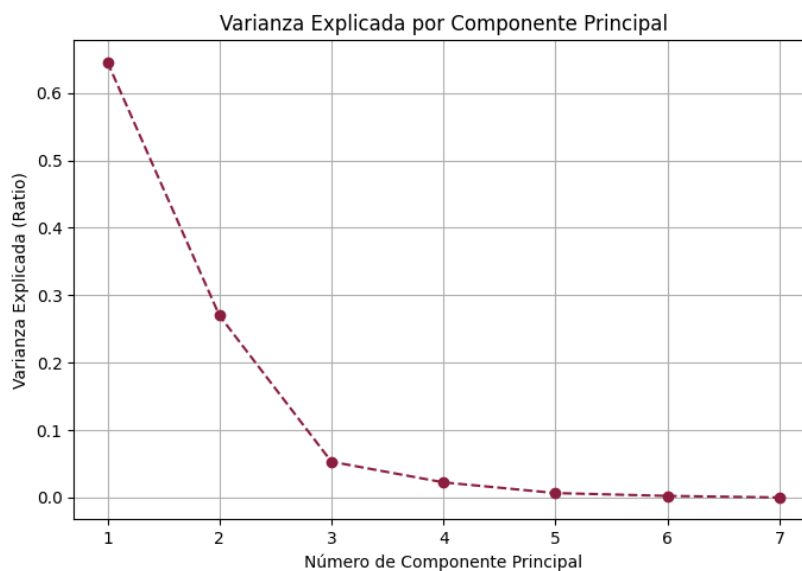
```
ax[0].set_title('Varianza Explicada por Componente Principal')
ax[0].set_xlabel('Número de Componente Principal')
ax[0].set_ylabel('Varianza Explicada (Ratio)')
ax[0].grid(True)

ax[1].plot(range(1, len(evr) + 1), np.cumsum(evr), marker='o', linestyle='-', color=BLUE)
ax[1].set_title('Varianza Explicada Acumulada')
ax[1].set_xlabel('Número de Componente Principal')
ax[1].set_ylabel('Varianza Explicada Acumulada (Ratio)')
ax[1].grid(True)

plt.tight_layout()
plt.show()
```

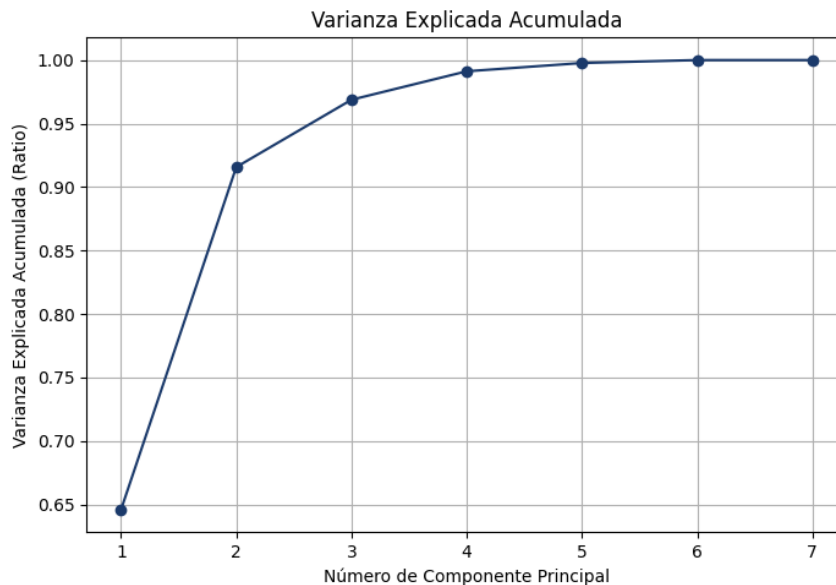
La función `subplots(1, 2)` genera una figura con dos paneles horizontales. El panel izquierdo representa la contribución individual de cada componente a la varianza total; el derecho, construido mediante `np.cumsum()`, acumula dichas proporciones. La instrucción `tight_layout()` ajusta automáticamente los márgenes para evitar superposiciones entre los elementos gráficos.

Ilustración 14 Varianza explicada por cada componente principal



Nota. El primer componente concentra aproximadamente el 65 % de la varianza y el segunda cerca del 27%

Ilustración 15 Varianza explicada acumulada de los componentes principales



Nota. Con dos componentes se conserva aproximadamente el 92 % de la variabilidad total.

12.12.4 Representación gráfica de las cargas factoriales

```
fig = plt.figure(figsize=(10, 6))

sns.heatmap(loadings, cmap='viridis', annot=True, fmt=".2f", linewidths=.5)

plt.title(f'Loadings de las Variables Originales en los {n_comp} Componentes Principales')

plt.xlabel('Componente Principal')

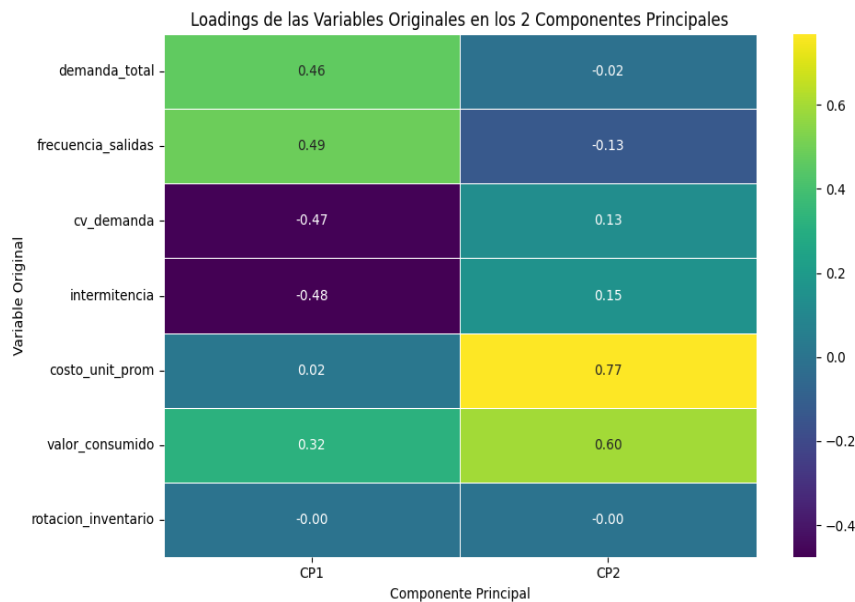
plt.ylabel('Variable Original')

plt.tight_layout()

plt.show()
```

La función `heatmap()` de Seaborn representa la matriz de cargas mediante una escala cromática continua. El parámetro `annot=True` superpone el valor numérico en cada celda, con formato de dos decimales según especifica `fmt`, lo que permite la lectura precisa de las cargas sin recurrir exclusivamente a la interpretación del color.

Ilustración 16 Cargas factoriales de las variables originales en los dos componentes principales



Nota. El primer componente se asocia al volumen y regularidad de la demanda; el segundo, al valor económico unitario.

12.12.5 Celda de registro de artefactos

```
# @title step_artifacts

num_fig = "2" # @param {type:"string"}

step = 'ModelOptimization' # @param ["DataLoading", "DataExploration", ...]

# upload_plt_to_gcs(num_fig, step, fig)
```

Esta celda corresponde a una plantilla de registro de artefactos propia del entorno Colab, empleada para catalogar las figuras generadas según la etapa metodológica a la que pertenecen. Las anotaciones @title y @param configuran controles de formulario en la interfaz. La instrucción de carga hacia el almacenamiento en la nube permanece comentada, por cuanto la función invocada no se encuentra definida en el entorno; en consecuencia, esta celda no ejecuta operación alguna sobre los datos y cumple una función exclusivamente documental.

12.13 Paso 6: Segmentación mediante K-means

12.13.1 Evaluación iterativa del número de conglomerados

```
#Selección de k (codo + silueta) y ajuste final

ks = range(2,9)

inertias, sils = [], []

for k in ks:

    kmk = KMeans(n_clusters=k, random_state=42, n_init=20).fit(scores_train)

    inertias.append(kmk.inertia_)

    sils.append(silhouette_score(scores_train, kmk.labels_))
```


El ciclo ajusta el algoritmo para cada valor de k comprendido entre dos y ocho, almacenando en cada iteración la inercia resultante —atributo `inertia_`, correspondiente a la suma de distancias cuadráticas intracluster— y el coeficiente de silueta promedio. El parámetro `n_init=20` determina que cada ajuste se ejecute veinte veces con inicializaciones aleatorias distintas de los centroides, conservando la solución de menor inercia; este procedimiento reduce sustancialmente el riesgo de convergencia hacia un óptimo local, vulnerabilidad reconocida del algoritmo (Arthur & Vassilvitskii, 2007).

12.13.2 Ajuste del modelo definitivo

```
K_ELEGIDO = 4

kmeans_final = KMeans(n_clusters=K_ELEGIDO, random_state=42, n_init=20).fit(scores_train)

labels_train = kmeans_final.labels_

labels_test = kmeans_final.predict(scores_test) # asignación por centroide más cercano
```

Con base en los criterios del codo y de la silueta se establece en cuatro el número de conglomerados. El modelo definitivo se ajusta exclusivamente sobre el conjunto de entrenamiento; el atributo `labels_` recupera la asignación resultante. Para el conjunto de prueba se emplea el método `predict()`, el cual asigna cada observación al centroide más próximo previamente determinado, sin ejecutar un nuevo entrenamiento. Esta distinción entre ajuste y predicción constituye el mecanismo que permite evaluar la capacidad de generalización del modelo.

12.13.3 Representación gráfica de los criterios de selección

```
fig, ax = plt.subplots(1, 2, figsize=(14, 5))

ax[0].plot(ks, inertias, marker='o', color=MAROON)

ax[0].set_title('Método del Codo (Inertia)')

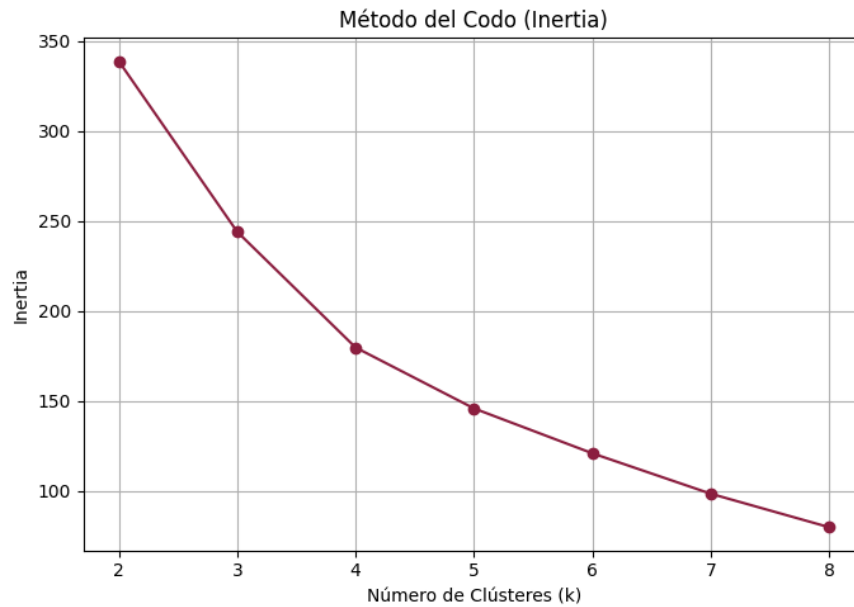
ax[0].set_xlabel('Número de Clústeres (k)')
```

```
ax[0].set_ylabel('Inertia')
ax[0].grid(True)
ax[1].plot(ks, sils, marker='o', color=BLUE)
ax[1].set_title('Coeficiente de Silueta')
ax[1].set_xlabel('Número de Clústeres (k)')
ax[1].set_ylabel('Puntuación de Silueta')
ax[1].grid(True)

plt.tight_layout()
plt.show()
```

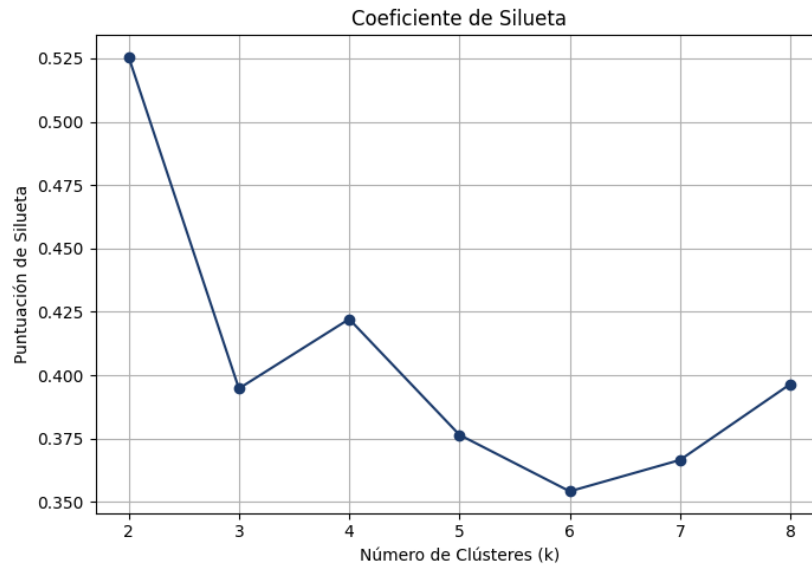
Las listas construidas en el bloque 6.1 se representan gráficamente en dos paneles: el izquierdo despliega la evolución de la inercia y el derecho la del coeficiente de silueta, ambos en función del número de conglomerados evaluado.

Ilustración 17 Inercia del algoritmo K-means para valores de k entre 2 y 8



Nota. El cambio de pendiente se observa en torno a $k = 4$.

Ilustración 18 Coeficiente de silueta promedio para valores de k entre 2 y 8



Nota. Se identifica un máximo local en $k = 4$ dentro del rango con granularidad suficiente para la segmentación.

La lectura conjunta de ambos criterios sustenta la elección de cuatro conglomerados: la curva de inercia evidencia una atenuación de las reducciones marginales a partir de dicho valor en tanto que el coeficiente de silueta registra un máximo local en el mismo punto, superando a las configuraciones de tres y cinco conglomerados (Rousseeuw, 1987). Si bien el valor máximo absoluto de silueta corresponde a $k = 2$, dicha configuración resulta excesivamente agregada para diferenciar políticas de reposición.

12.14 Paso 7: Evaluación y Perfilamiento

12.14.1 Métricas de validación interna

```
#Evaluación train vs. test  
sil_train = silhouette_score(scores_train, labels_train)  
sil_test  = silhouette_score(scores_test, labels_test)  
db_train  = davies_bouldin_score(scores_train, labels_train)  
db_test   = davies_bouldin_score(scores_test, labels_test)
```

Se calculan de forma paralela dos métricas de validación interna sobre ambos subconjuntos. El coeficiente de silueta estudia al mismo tiempo la cohesión interna y la división entre conglomerados, con valores próximos a la unidad indicando asignaciones adecuadas (Rousseeuw, 1986).

El índice de Davies-Bouldin cuantifica la semejanza promedio entre cada conglomerado y su vecino más semejante, siendo preferibles los valores reducidos (Bouldin, 1979). La comparación entre entrenamiento y prueba permite verificar que la estructura identificada no depende exclusivamente de la muestra empleada para el ajuste.

12.14.2 Consolidación de resultados y perfilamiento

```
#Perfil de clústeres

feat_train['cluster'], feat_train['split'] = labels_train, 'train'

feat_test['cluster'], feat_test['split'] = labels_test, 'test'

resultado = pd.concat([feat_train, feat_test], ignore_index=True)


perfil = resultado.groupby('cluster')[vars8].mean()

perfil['n_productos'] = resultado.groupby('cluster').size()
```

Las etiquetas de conglomerado se incorporan a los conjuntos de características originales, añadiendo asimismo un indicador de procedencia que permite distinguir posteriormente el origen de cada observación. La función `concat()` une ambos subconjuntos en una sola tabla. El perfilamiento se logra agrupando por etiqueta de conglomerado y calculando el promedio de las variables en sus unidades originales (sin transformación logarítmica ni estandarización), lo que garantiza la interpretabilidad de los resultados en términos operativos. Se incorpora finalmente el conteo de productos por segmento. Esta etapa reviste importancia equivalente a la generación técnica de los conglomerados, por cuanto una segmentación que no admita traducción en políticas diferenciadas carece de utilidad práctica (Hunter, 2007).

12.14.3 Presentación del perfil de conglomerados

```
# Mostrar el perfil de clústeres

print("Perfil de Clústeres (valores promedio de las variables originales y número de productos por clúster):")

print(perfil)
```

	demanda_total	frecuencia_salidas	cv_demanda	intermitencia	\
clúster					
0	1539.619048	5.976190	2.788536	0.835238	
1	859.112000	3.349398	3.301784	0.889639	
2	26368.000000	336.500000	0.375584	0.000000	
3	21211.785714	300.404762	0.672913	0.157143	

	costo_unit_prom	valor_consumido	saldo_promedio	\
clúster				
0	49.135890	4.729461e+04	NaN	
1	3.315500	1.884048e+03	NaN	
2	9452.000000	3.132723e+08	NaN	
3	3.013286	5.478701e+04	NaN	

	rotacion_inventario	n_productos
clúster		
0	0.0	42
1	0.0	83
2	0.0	2
3	0.0	42

La salida presenta la caracterización cuantitativa de los cuatro segmentos identificados. Se observa una estructura marcadamente asimétrica: el conglomerado 2, integrado por únicamente dos productos, concentra el mayor valor consumido; el conglomerado 3 agrupa 42 artículos de alta rotación y bajo costo unitario; y los conglomerados 0 y 1 reúnen conjuntamente 125 productos caracterizados por intermitencias superiores al 80 %. La columna de saldo promedio registra valores indeterminados y la de rotación valores nulos, consecuencia de la limitación en la fuente de datos previamente señalada.

12.14.4 Visualización de los conglomerados

```
plt.figure(figsize=(10, 8))

centroids = kmeans_final.cluster_centers_

for i in range(K_ELEGIDO):

    plt.scatter(scores_train[labels_train == i, 0],

                scores_train[labels_train == i, 1],

                label=f'Clúster {i}', alpha=0.7, s=50,

                c=PALETTE[i % len(PALETTE)])

plt.scatter(centroids[:, 0], centroids[:, 1],

            marker='X', s=200, color='black', edgecolor='white',

            label='Centroides', zorder=2)

plt.title('Clústeres de Productos en el Espacio de Componentes Principales')

plt.xlabel('Componente Principal 1')

plt.ylabel('Componente Principal 2')

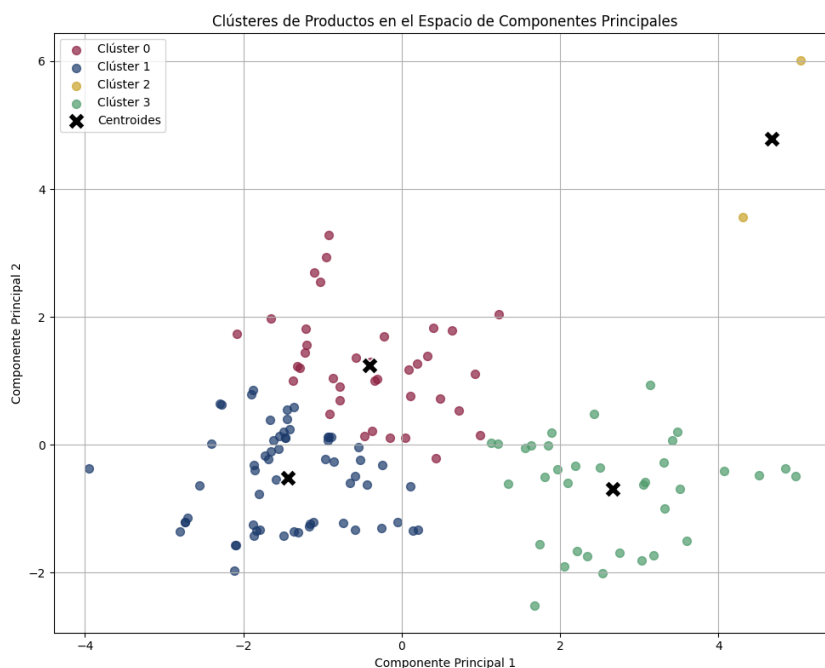
plt.legend()

plt.grid(True)
```

El ciclo genera un diagrama de dispersión independiente para cada conglomerado, filtrando las observaciones mediante registro booleano sobre el vector de etiquetas y aplicando el color correspondiente de la paleta definida en la fase inicial. El parámetro alpha establece la transparencia

de los marcadores, facilitando la visualización de zonas de alta densidad. Una segunda invocación superpone los centroides con marcador distintivo, tamaño aumentado y parámetro `zorder=2`, que asegura su representación en primer plano.

Ilustración 19 Conglomerados de productos en el espacio de componentes principales.



Nota. Las marcas en forma de X corresponden a los centroides finales del modelo.

12.14.5 Celda final de registro de artefactos

```
# @title step_artifacts

num_fig = "4" # @param {type:"string"}

step = 'DataWrangling' # @param ["DataLoading", "DataExploration", ...]

# upload_plt_to_gcs(num_fig, step, fig)
```

La celda final replica la plantilla de registro documental descrita anteriormente, catalogando la cuarta figura del cuaderno. Al igual que sus homólogos, no ejecuta operaciones sobre los datos ni modifica los objetos en memoria, por lo que su omisión no altera los resultados del modelo. Con esta instrucción concluye la ejecución del script.

12.15 Interpretación de los Resultados

El perfilamiento del conjunto, obtenido mediante el promedio de las variables originales agrupadas por etiqueta, permite caracterizar operativamente cada segmento identificado y traducir los hallazgos estadísticos en criterios de gestión aplicables al inventario de la empresa.

Tabla 8: Perfil promedio de los conglomerados obtenidos. Fuentes: Elaboración propia.

Clúster	Demanda total	Frecuencia de salidas	CV demanda	Intermitencia	Costo unitario	Productos
0	1 539.62	5.98	2.79	0.835	49.14	42
1	859.11	3.35	3.30	0.890	3.32	83
2	26 368.00	336.50	0.38	0.000	9 452	2
3	21 211.79	300.40	0.67	0.157	3.01	42

Nota. Valores promedio de las variables originales sin transformar. La variable rotación de inventario se excluye de la tabla por registrar valores nulos derivados de la ausencia de datos en el campo de saldo del kárdex.

12.15.1 Conglomerado 2: productos críticos de alto valor

Este segmento, conformado únicamente por dos productos, registra la demanda total más elevada del portafolio (26 368 unidades en promedio), la mayor frecuencia de despacho (336.5 salidas), el coeficiente de variación más bajo (0.376) y una intermitencia nula, lo que demuestra un consumo estable y altamente predecible. Su costo unitario promedio (USD 9 452) resulta superior al del resto de segmentos. La combinación de demanda estable y alto valor económico configura productos de criticidad máxima para la operación, cuya gestión admite modelos determinísticos de reposición con niveles reducidos de inventario de seguridad, dada la baja incertidumbre asociada a su demanda.

12.15.2 Conglomerado 3: productos de rotación masiva

Integrado por 42 productos, este segmento presenta una demanda total promedio de 21 211.79 unidades y una frecuencia de salidas de 300.40, acompañadas de un coeficiente de variación moderado (0.673) y una intermitencia reducida (15.7 %). Su costo unitario promedio (USD 3.01) es el más bajo del portafolio. Se trata, por tanto, de artículos de alta rotación y bajo valor individual, cuya relevancia radica en el volumen y en su incidencia sobre el nivel de servicio. Su gestión es compatible con políticas de revisión periódica y reposición automatizada.

12.15.3 Conglomerado 0: productos de demanda intermitente y valor intermedio

Conformado por 42 productos, en este segmento se registra una demanda total promedio de 1 539.62 unidades, una frecuencia de salidas considerablemente menor (5.98) y un coeficiente de variación elevado (2.789), con una intermitencia del 83.5 %. Su costo unitario promedio (USD 49.14) lo sitúa dentro de un rango intermedio. La conjunción de baja frecuencia, alta variabilidad y valor unitario no despreciable configura el escenario de mayor dificultad para la planificación, por cuanto tanto la escasez como el exceso de inventario generan costos relevantes.

12.15.4 Conglomerado 1: productos de baja rotación y bajo valor

Constituye el segmento más numeroso del portafolio, con 83 productos. Presenta la demanda total promedio más reducida (859.11 unidades), la menor frecuencia de despacho (3.35), el coeficiente de variación más elevado (3.302) y la intermitencia más alta (89.0 %), junto con un costo unitario promedio de USD 3.32. Estos artículos concentran el mayor riesgo de inmovilización de inventario, dado que su consumo resulta esporádico y su valor consumido es marginal, situación que sugiere la conveniencia de evaluar su permanencia en el portafolio o de establecer políticas de reposición bajo pedido.

12.16 Consideraciones generales

Los resultados obtenidos evidencian una estructura de portafolio marcadamente asimétrica: dos productos concentran el mayor valor consumido, mientras que 125 artículos (correspondientes a los conglomerados 0 y 1) presentan intermitencias superiores al 80 %. Esta configuración confirma la pertinencia de establecer políticas de reposición diferenciadas por segmento, en línea con lo señalado por (Huber, 2017), quien sostiene que la caracterización de los conglomerados posee tanta relevancia como su generación técnica, por cuanto una segmentación que no se traduzca en decisiones operativas diferenciadas carece de utilidad práctica.

Finalmente, corresponde señalar como limitación del estudio que la variable rotación de inventario no pudo incorporarse efectivamente al modelo debido a la ausencia de registros válidos en el campo de saldo del kárdex, circunstancia que restringe el análisis a las dimensiones de volumen, regularidad y valor económico. Se recomienda, para investigaciones posteriores, subsanar dicha deficiencia en la fuente de datos a fin de incorporar la dimensión de eficiencia en el uso del inventario.

13 DISCUSIÓN

Este trabajo de investigación se basa en emplear el algoritmo K-Means, junto con el Análisis de Componentes Principales (ACP), con la finalidad de identificar mediante indicadores de demanda, costo, saldo y rotación el comportamiento de los productos dentro del inventario, para así, reconocer qué grupos de productos requieren un manejo diferente dentro del proceso de optimización del stock. Este análisis se puede contrastar con el proyecto "***Demand-Based Product Classification Using K-Means with Intermittency Metrics***" (Ariska Nur Anggraini, 2026), el cual identifica mediante el mismo algoritmo, patrones de demanda intermitente en un sistema de inventario multi-SKU, con el fin de formular estrategias de aprovisionamiento diferenciadas por clúster. En este trabajo, se recopilaron datos de la empresa PT JJA, de transacciones registradas

En este trabajo se recopilaron datos directamente del objeto de estudio de la empresa PT JJA, a partir de 11,988 transacciones registradas entre los años 2020 y 2025, periodo que incluye la contingencia sanitaria por COVID-19, se procesaron esas transacciones hasta obtener 261 productos, cada uno representado por su historial de ventas mensuales a lo largo de 72 periodos.

Al iniciar, los autores realizaron la limpieza y depuración de los datos, verificando que no existieran valores perdidos y registrando como cero los meses en los que no hubo transacciones para un producto determinado, con el fin de preservar la estructura temporal completa de la demanda. Posteriormente, estandarizaron y normalizaron las variables mediante z-score, debido a la sensibilidad del algoritmo K-Means a la escala de las variables —el mismo procedimiento de estandarización empleado en la presente investigación.

Dada la complejidad de las variables relacionadas con la demanda intermitente, los autores construyeron seis atributos derivados por producto: la demanda total, la demanda mensual promedio, el coeficiente de variación (CV), el número de meses activos, la proporción de meses sin ventas

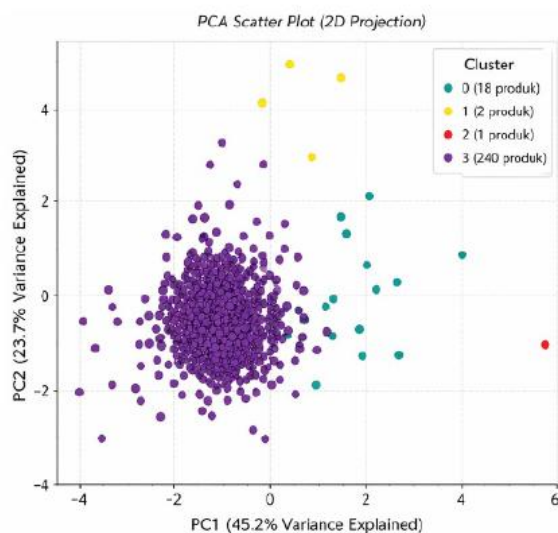
(zero_month_ratio) y el intervalo promedio de demanda (ADI, Average Demand Interval), además de la tendencia de la serie. Estos indicadores son conceptualmente equivalentes a los empleados en la presente investigación —demanda, frecuencia de salidas, coeficiente de variación e intermitencia— lo que confirma que ambos estudios recurren a un conjunto de variables similar para caracterizar el comportamiento del inventario antes de aplicar el agrupamiento.

El modelo empleado corresponde a la función objetivo estándar del algoritmo K-Means, la cual minimiza la varianza intra-clúster mediante optimización iterativa basada en centroides:

$$J = \sum_{i=1}^k \sum_{x \in C_i} ||x - \mu_i||^2$$

Para determinar el número óptimo de clústeres, los autores combinaron el método del codo (evaluando el descenso de la suma de errores cuadráticos, SSE) con el coeficiente de silueta, evaluado en el rango $k = 2$ a $k = 10$, con inicialización k-means++ y un máximo de 300 iteraciones —procedimiento metodológicamente idéntico al empleado en el presente estudio.

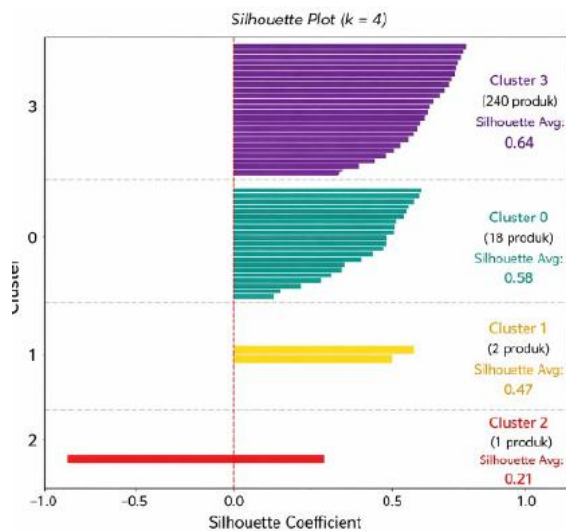
Ilustración 20 Proyección PCA de la separación de clústeres obtenida en Anggraini, Amali y Anwar (2026)



Nota: El primer componente explica el 45.2 % de la varianza y el segundo el 23.7

El siguiente resultado nos permite evaluar que, en este caso, el modelo de K-Means alcanzó su mejor desempeño con $k = 4$, con un coeficiente de silueta de 0.62, superior al obtenido con $k = 3$ (≈ 0.54) y con $k = 5$ (≈ 0.58). La distribución final de los 261 productos fue marcadamente desbalanceada: el Clúster 3 concentró 240 productos, mientras que los Clústeres 0, 1 y 2 agruparon únicamente 18, 2 y 1 productos, respectivamente. Esta estructura de "cola larga" (long-tail) es, según los propios autores, una característica natural de los sistemas de inventario multi-SKU, donde la mayoría de los productos presenta un patrón de demanda intermedio, mientras que un número reducido de productos exhibe comportamientos extremos —ya sea de altísima variabilidad o de demanda prácticamente inexistente.

Ilustración 21 Ejemplos de resultados de los productos clusterizados.



Nota: Los productos del Clúster 3 presentan valores de silueta positivos y altos, indicando una buena calidad de agrupamiento, mientras que el clúster pequeño mantiene una separación válida y consistente. Obtenida en Anggraini, Amali y Anwar (2026).

En base a la caracterización de los grupos extremos, el Clúster 2 —conformado por un solo producto— presentó el valor de ADI mayor (18.44) y el ZMR más alto (0.96), lo que demuestra un patrón de demanda extraño, con intervalos largos entre transacciones y una gran proporción de meses sin ventas. El Clúster 1, con solo dos productos, registró el CV más alto (4.85), demostrando cambios

de demanda extremas e impredecibles. El Clúster 0, con 18 productos, demostró valores intermedios de ADI (9.87) y CV (3.12), pero un ZMR considerablemente alto (0.91). Esta diferenciación confirma que cada clúster captura un patrón estructuralmente diferente de comportamiento temporal de la demanda, y que dicho comportamiento no puede gestionarse mediante una política de aprovisionamiento igual.

Tabla 9: Análisis de silueta de los clústeres obtenidos. Fuentes: Anggraini, Amali y Anwar (2026).

Product Code	Item Name	Cluster	Cluster Type	Procurement Strategy
AA-1510	HONEYWELL EDA 52	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1520	Water Wind Glue Shot Caulking Gun	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1530	CBA-R07-S07PAR Cable RS232	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1540	DS3678-SR Rugged Scanner Kit	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1550	310P1.25 NAC Impact Sockets	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1560	Makita GA 4030	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1570	Tube Package APAR Tonata ABC 1 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1580	Tube Package APAR Tonata HFC 6 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1590	Tube Package APAR Tonata HFC 1 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring

Notas. Ejemplos de productos del Clúster 3 y sus estrategias de abastecimiento, utilizados como apoyo para el análisis de demanda y la gestión de inventarios. Obtenida en Anggraini, Amali y Anwar (2026).

Ambas investigaciones se centran en la optimización del inventario a partir de técnicas de agrupamiento no supervisado; sin embargo, se tienen premisas similares las cuales conllevan a similitudes estructurales, como lo son el uso del algoritmo K-Means como herramienta central de segmentación, el empleo del coeficiente de variación y de la intermitencia como variables clave, y la validación de los clústeres mediante el coeficiente de silueta.

Adicional a la dimensión temporal de la demanda discutida, resulta relevante también evaluar los aspectos metodológicos propios de la arquitectura ACP-K-Means empleada en la presente investigación, particularmente en lo referente a la selección de sus hiperparámetros. En este sentido,

el proyecto puede contrastarse también con la investigación de (Jackson, 2022), "AutoML Approach to Stock Keeping Units Segmentation", la cual combina ACP y K-Means para segmentar unidades de mantenimiento de existencias (SKU), con el fin de superar las limitaciones del método clásico de clasificación ABC.

Se recopilaron datos directamente del objeto de estudio, un proveedor logístico de terceros de la región del Báltico, a partir de un conjunto de datos real a nivel de pallet, utilizó un conjunto de 1,218 filas correspondientes a SKU individuales, caracterizadas por cuatro atributos: precio unitario, número de unidades por pallet, peso bruto del pallet cargado y altura del pallet.

Como parte de la metodología el autor verificó la presencia de multicolinealidad entre las variables mediante el coeficiente de correlación de Pearson, encontrando una correlación moderada entre el peso bruto y el precio unitario ($r = 0.61$), lo que sugirió que el ACP tenía potencial para mejorar los resultados del agrupamiento al reducir la redundancia entre variables. Posteriormente, a diferencia de la estandarización z empleada en la presente investigación, el autor aplicó un escalado min-máx para llevar todas las variables al rango $[0, 1]$, dado que K-Means es sensible a la escala de las variables:

$$x = \frac{x - x_{min}}{x_{max} - x_{min}}$$

El autor construyó un pipeline de cuatro etapas (debido a la complejidad de las variables relacionadas con las características tanto físicas como comerciales del SKU): almacenamiento y recuperación de datos, preprocesamiento y estandarización, reducción de dimensionalidad mediante ACP, y agrupamiento mediante K-Means. El modelo de agrupamiento empleado corresponde a una variante computacionalmente más eficiente del algoritmo estándar, denominada minibatch K-Means, la cual minimiza la misma función objetivo mediante el uso de submuestras (minibatches) actualizadas

interactivamente:

$$J = \sum_{i=1}^k \sum_{x \in C_i} ||x - \mu_i||^2$$

Ilustración 22 Algoritmo MiniBatch K-means

Algorithm 1 Minibatch K-means [37].

```

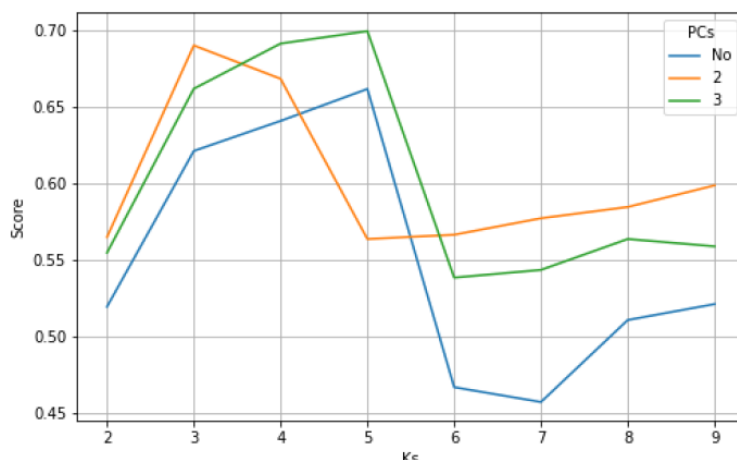
1: Given:  $k$ , minibatch size, iterations  $t$ , data set  $X$ 
2: Initialize each  $c \in C$  with an  $x$  picked randomly from  $X$ 
3:  $v \leftarrow 0$ 
4: for  $i = 1$  to  $t$  do
5:    $M \leftarrow$  size examples picked randomly from  $x$ 
6:   for  $x \in M$  do
7:      $d[x] \leftarrow f(C, x)$  ▷ Store the center nearest to  $x$ 
8:   end for
9:   for  $x \in M$  do
10:     $x \leftarrow d[x]$  ▷ Get cached center for this  $x$ 
11:     $v[c] \leftarrow v[c] + 1$  ▷ Update per-center counts
12:     $\eta \leftarrow \frac{1}{v[c]}$  ▷ Get per-center learning rate
13:     $c \leftarrow (1 - \eta)c + \eta x$  ▷ Take gradient step
14:   end for
15: end for

```

Notas. Utilizado para la agrupación eficiente de datos mediante la actualización iterativa de centroides a partir de mini lotes de observaciones. Obtenido de AutoML Approach to Stock Keeping Units Segmentation.

Para la reducción de dimensionalidad, el ACP se formuló de forma idéntica a la empleada en la presente investigación: estandarización de la matriz de datos, cálculo de la matriz de covarianza y diagonalización de dicha matriz para obtener los autovalores y autovectores, estos últimos ordenados de forma descendente según la varianza que explican. La validación de los clústeres se realizó mediante el coeficiente de silueta, calculado a partir de la cohesión y la separación de cada observación a diferencia de la presente investigación, en la que el número de componentes y de clústeres se determina mediante el método del codo, la silueta y el índice de Davies-Bouldin evaluados de forma manual, el aporte principal del artículo de (Jackson, 2022) es automatizar por completo esa selección mediante una búsqueda en cuadrícula (grid search): el autor evaluó de forma exhaustiva el producto cartesiano entre el número de clústeres ($K = 2$ a 9) y el número de componentes principales ($P = 0, 2$ o 3 , donde 0 corresponde a no aplicar ACP), reteniendo la combinación con el mayor coeficiente de silueta.

Ilustración 23 Coeficientes de silueta



Nota. La combinación de 3 componentes principales y 5 clústeres obtuvo el mejor desempeño (silueta = 0.699).

El siguiente resultado nos permite evaluar que, en este caso, el modelo con tres componentes principales y cinco clústeres obtuvo el mejor desempeño, con un coeficiente de silueta de 0.699. Es de suma importancia para la presente investigación que, incluso sin aplicar ACP, el óptimo también se alcanzó con cinco clústeres, pero con un desempeño menor (silueta = 0.662); mientras que la combinación de dos componentes principales y tres clústeres obtuvo el segundo mejor resultado (silueta = 0.690). Estos hallazgos confirman que la incorporación del ACP antes del agrupamiento K-Means mejora la calidad de la segmentación, resultado que respalda directamente la decisión metodológica de la presente investigación de aplicar ACP como etapa previa al agrupamiento.

Por otro lado, el objetivo de este estudio es identificar correctamente la relación entre las variables de demanda, frecuencia de salidas, coeficiente de variación, intermitencia, costo unitario, valor consumido, saldo promedio y rotación de inventario, con enfoque en segmentar los productos del inventario según su comportamiento de movimiento y así optimizar el nivel de stock necesario para cada grupo. En contraste, el objetivo de la investigación de (Jackson, 2022), es automatizar la selección de los hiperparámetros del pipeline ACP-K-Means (el número de clústeres y el número de componentes principales) para reducir el esfuerzo manual de calibración en aplicaciones reales de

segmentación de SKU. Estos proyectos se centran en la optimización del inventario a partir de la misma arquitectura metodológica de dos etapas (ACP + K-Means); sin embargo, se tienen premisas similares las cuales conllevan a similitudes estructurales, como lo son el uso de la estandarización previa de variables, la validación de los clústeres mediante el coeficiente de silueta, y la crítica compartida al método clásico de clasificación ABC por considerar una única dimensión del comportamiento del inventario.

El estudio de (Jackson, 2022), se centró en demostrar que el método ABC tradicional resulta insuficiente porque dos SKU pueden tener una contribución similar a los ingresos, pero comportamientos completamente. En consecuencia, el autor logró demostrar que un pipeline automatizado de ACP y K-Means puede superar sistemáticamente al ABC clásico al considerar múltiples dimensiones simultáneamente, y que la automatización de la calibración de hiperparámetros (tradicionalmente un proceso manual, tedioso y dependiente de especialistas) puede resolverse mediante búsqueda en cuadrícula guiada por el coeficiente de silueta.

En contraste, la presente investigación, la cual busca identificar no solo el comportamiento físico o el precio de los productos, sino también la dinámica de su demanda y su impacto financiero en el inventario (combinando demanda, frecuencia, variabilidad, costo, valor consumido, saldo y rotación), busca identificar a su vez un perfil de comportamiento comercial más completo que el basado en atributos físicos del pallet empleado por Jackson. Con la finalidad de poder crear una segmentación orientada específicamente a la gestión de existencias desde una perspectiva de demanda y costo, la cual servirá para diseñar políticas de reabastecimiento diferenciadas según el tipo de producto, complementando así el enfoque de automatización de hiperparámetros propuesto por el artículo de referencia con una selección de variables más orientada al comportamiento comercial del inventario.

14 CONCLUSIONES

Este trabajo de investigación demuestra la importancia del análisis de datos y el uso del concepto de Inteligencia Artificial como lo es el Machine Learning en la gestión de inventarios, especialmente en la segmentación de productos en función de su demanda, costo y rotación. Emplear estas metodologías proporciona una mejor calidad de la información, gracias a ello, se pudo determinar que productos requieren prioridad y cuales pueden manejarse con diferentes tipos de reabastecimiento, los cuales podrían llegar a ser más económicos. Logrando alcanzar mayor eficiencia en la administración del stock dentro de la empresa.

Mediante la revisión literaria del comportamiento de la demanda en sistemas de inventario y de los algoritmos de Machine Learning que sean aptos para este tipo de análisis, se logró demostrar de forma sencilla cómo el Análisis de Componentes Principales y el algoritmo K-Means pueden aplicarse en el sector logístico y por qué son tan viables para estudios como este, al permitir reducir la dimensionalidad de múltiples indicadores sin perder información relevante del comportamiento del producto.

Se entrenaron modelos de Machine Learning con datos históricos de kárdex de la empresa TOALCO, correspondientes al periodo 2023-2025. Con este método se pudo aislar la información de demanda, frecuencia de salidas, variabilidad, costo unitario, valor consumido, saldo promedio y rotación de inventario, y así poder encontrar diferencias en el comportamiento entre los distintos grupos de productos. Esta clasificación permitió conocer cómo afecta cada variable en los resultados de gestión, pues no todos los productos presentan el mismo patrón de movimiento ni requieren la misma estrategia de aprovisionamiento.

Los resultados se compararon con otros estudios de metodología similar, como los de (Ariska Nur Anggraini, 2026) y (Jackson, 2022), coincidiendo en que la variabilidad de la demanda y la reducción de dimensionalidad mediante ACP son determinantes para obtener una segmentación de inventario más robusta que la ofrecida por métodos tradicionales de una sola dimensión, como el clásico ABC. Esto validó los hallazgos y demostró que las herramientas de análisis de datos no supervisado se pueden usar de forma efectiva en la gestión de inventarios.

En conclusión, la segmentación de productos de inventario mediante modelos como ACP y K-Means es una herramienta que puede ayudar a mejorar y optimizar la gestión del stock dentro de la empresa. Una buena interpretación de los datos ayuda a las organizaciones a tomar mejores decisiones, como también a diseñar estrategias de abastecimiento diferenciadas por grupo de productos, y a disminuir riesgos de sobre stock y desabastecimiento. Es de este modo como el uso del Machine Learning y el análisis estadístico contribuyen a alcanzar los objetivos de esta investigación.

15 REFERENCIAS

- Angulo-Rivera, R. J. (2022). *Control interno y gestión de inventarios de la empresa constructora Peter Contratistas S.R.* Obtenido de <https://revistas.unheval.edu.pe/index.php/gacien/article/view/696>
- Aris A. Syntetos, Z. B. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *scienceDirect*.
- Ariska Nur Anggraini, A. A. (2026). *research gate*. Obtenido de https://www.researchgate.net/publication/404513136_Demand-Based_Product_Classification_Using_K-Means_with_Intermittency_Metrics
- Ballou, R. H. (2004). *Logística- Administración de la cadena de suministro*. Pearson.
- Banco Central del Ecuador. (2025). Obtenido de <https://www.bce.fin.ec/la-economia-ecuatoriana-crecio-24-en-el-tercer-trimestre-de-2025/>
- BancoMundial. (s.f.). *Banco Mundial*. Obtenido de <https://www.bancomundial.org/es/topic/urbandevelopment/overview>
- Barto, R. S. (2018). *Reinforcement Learning*. The MIT Press.
- Bell, R. (noviembre de 2025). *oracle*. Obtenido de oracle: <https://www.oracle.com/latam/construction-engineering/ai-construction/>
- Bhandari, A. (2026). *Construction Materials Market Size, Share & Industry Analysis, By Material Type (Aggregates, Cement, Bricks and Blocks, Metals, and Others), By End-user (Residential, Infrastructure, Commercial, and Industrial), and Regional Forecast, 2026-2034*.
- Bouldin, D. (1979). *datasciencepythonblog*. Obtenido de <https://datasciencepythonblog.net/davies-bouldin-score-scikit-learn/>
- Carbonneau, R. (2011). *Application of machine learning techniques for supply chain demand forecasting*. ELSEIVER.
- Chen, M. (2024). Obtenido de <https://www.ibm.com/es-es/think/topics/unsupervised-learning>
- China, C. R. (15 de OCTUBRE de 2021). Obtenido de <https://www.ibm.com/mx-es/think/topics/machine-learning-types>
- Cuervo, F. G. (2021). *Inteligencia Artificial (IA): Una nueva oportunidad para el sector de la construcción*. EY | Building a better working world.
- David Arthur, S. V. (2007). *k-means: The Advantages of Careful Seeding*.
- Domingos, P. (2012). *A few useful things to know about machine learning*. CommuniCAtionS oF thE ACm.
- Edmonds, G. A. (1979). The construction industry. 15.
- Edward A. Silver, D. F. (2016). *Inventory and Production Management in Supply Chains*.
- Ellis, G. (Marzo de 2026). *Autodesk*. Obtenido de Autodesk: <https://www.autodesk.com/blogs/construction/ai-construction/>
- Foster Provost, T. F. (2013). *Data Science for Business*. Obtenido de https://www.researchgate.net/publication/256438799_Data_Science_for_Business
- Garrido , Cejas. (2017). LA GESTIÓN DE INVENTARIO COMO FACTOR ESTRATÉGICO EN LA.
- Giovanna Culot, M. P. (noviembre de 2024). *Artificial intelligence in supply chain management: A systematic literature review of empirical studies and research directions*. Obtenido de science direct: <https://www.sciencedirect.com/science/article/pii/S0166361524000605?via%3Dihub>

- Grané, A. (2002). *Análisis de Componentes Principales*. Universidad Carlos III de Madrid, Departamento de Estadística, Madrid.
- Harabasz, T. C. (1974). *A dendrite method for cluster analysis*. Communication statistic.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
- Hazen, B. T. (2016). *Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications*. Elseiver.
- Huber, J. (2017). Cluster-based hierarchical demand forecasting for perishable goods. *Science direct*.
- Hunter, J. D. (Mayo de 2007). *IEEE XPLORE*. Obtenido de <https://ieeexplore.ieee.org/document/4160265>
- Jaadi, Z. (s.f.). *IBM*. Obtenido de <https://www.ibm.com/es-es/think/topics/clustering>
- Jackson, I. (noviembre de 2022). *MDPI*. Obtenido de <https://www.mdpi.com/0718-1876/17/4/76>
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Science direct*.
- Jolliffe, T. (2002). *Principal Component Analysis*. Springer Series in Statistics (SSS).
- Lewotsky, K. (29 de MARZO de 2022). *ASSOCIATION FOR ADVANCING AUTOMATION*. Obtenido de ASSOCIATION FOR ADVANCING AUTOMATION: <https://www.automate.org/ai/industry-insights/how-unsupervised-machine-learning-benefits-industrial-automation>
- Ley Orgánica De Protección De Datos Personales, a. 1. (2021). Obtenido de Ley Orgánica De Protección De Datos Personales, art 10: https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf
- Lloyd, S. (1982). *Least squares quantization in PCM*. IEEE.
- Lolli, F. (2013). *New AHP-based approaches for multi-criteria inventory classification*. ELSEIVER.
- Luque, R. (2026). Obtenido de <https://plan-vivienda.mit.gob.ec/casa-miti-miti>
- M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela. (2012). A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm.
- M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela. (2012). *Cornell University*. Obtenido de <https://arxiv.org/abs/1209.1960>
- MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*.
- Meindl, P. C. (2008). *Administración de la cadena de suministro*. Pearson.
- Ministerio de Economía y Finanzas. (2026). Obtenido de <https://www.finanzas.gob.ec/politicas-del-gobierno-del-presidente-daniel-noboa-fortalecen-la-confianza-en-el-sector-de-la-construccion/>
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- Monreal, G. (2025). *Asociación española técnicos en cerámica*. Obtenido de <https://atece.org/ia-en-el-sector-ceramico/>
- Munárriz, L. Á. (1994). *FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL*. Murcia: edit.um.
- Navarro Céspedes, J. (2010). análisis de componentes principales y análisis de regresión para datos categóricos. *Revista de Matemática: Teoría y Aplicaciones*.
- Purcell, B. (s.f.). *Biz Tech Magazine*. Obtenido de https://biztechmagazine-com.translate.google/article/2025/05/what-are-benefits-unsupervised-machine-learning-and-clustering-perfcon?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Real Carbonneau, e. a. (1 de febrero de 2008). *ScienceDirect*. Obtenido de ScienceDirect: <https://www.sciencedirect.com/science/article/abs/pii/S0377221706012057?via%3Dihub>
- Rosicart, E. G. (11 de febrero de 2026). *OBS Bussines School*. Obtenido de OBS Bussines School: <https://www.obsbusiness.school/blog/el-aprendizaje-no-supervisado-y-como-descubrir-patrones-en-datos>

- Rousseeuw, P. J. (1986). *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*. ELSEVIER.
- Sharon, T. (2022). Obtenido de Universidad Técnica de Ambato:
<https://repositorio.uta.edu.ec/server/api/core/bitstreams/57857425-7f53-4006-b097-6eaa0c364c0d/content>
- Silver, E. A. (2016). *Inventory and Production Management in Supply Chains*. Taylor & Francis Group.
- Stuart Russell, P. N. (2008). *Inteligencia Artificial, Un Enfoque Moderno*. Pearson Education, Inc, publishing as Prentice Hall.
- Thorndike, R. L. (1953). *Who Belongs in the Family?* Cambridge.
- Trevor Hastie , Robert Tibshirani , Jerome Friedman. (2009). *The Elements of Statistical Learning*. Springer.
- Williams, L. J. (2010). *Principal component analysis*. Chaomei Chen.

16 ANEXOS

16.1 Anexo 1. Base de datos

codigo	Artículo	Línea	+	-	Saldo	costo Último	costo Promedio	Precio	valor
0.148	CEMENTO BLANCO HUASCARAN (50KG.)	MATERIALES	60	0	0				1,859.46
1	PORCELANA BLANCO 1K	EMPORE	50,170	18,015					10,000.00
10	PORCELANA MIEL 2K	EMPORE	81	671					0.00
101	POLVO MINERAL TIÑEMAX AMARILLO OCRE 25KG	POLVO MINERAL	0	45					0.00
102	POLVO MINERAL TIÑEMAX AMARILLO PATITO 25KG.	POLVO MINERAL	0	133					0.00
103	POLVO MINERAL TIÑEMAX NEGRO 25KG.	POLVO MINERAL	0	108					0.00
104	POLVO MINERAL TIÑEMAX ROJO SUAVE 25KG.	POLVO MINERAL	0	16					0.00
105	POLVO MINERAL TIÑEMAX CAFE 25KG.	POLVO MINERAL	0	13					0.00
106	POLVO MINERAL TIÑEMAX NARANJA 25KG.	POLVO MINERAL	0	0					0.00
107	POLVO MINERAL TIÑEMAX AZUL 25KG.	POLVO MINERAL	0	38					0.00
108	POLVO MINERAL TIÑEMAX AZUL FUERTE 25KG.	POLVO MINERAL	0	32					0.00
11	PORCELANA PALO ROSA 1K	EMPORE	0	560					0.00
111	POLVO MINERAL TIÑEMAX ROJO FUERTE 25KG.	POLVO MINERAL	0	18					0.00
112	POLVO MINERAL TIÑEMAX VERDE FUERTE 25KG.	POLVO MINERAL	0	19					0.00
113	POLVO MINERAL TIÑEMAX VERDE CLARO 25KG.	POLVO MINERAL	0	12					0.00
114	POLVO MINERAL TIÑEMAX VERDE OSCURO 25KG.	POLVO MINERAL	0	16					0.00
115	POLVO MINERAL TIÑEMAX BLANCO 25KG.	POLVO MINERAL	0	16					0.00
116	POLVO MINERAL TIÑEMAX CAOBA 25KG.	POLVO MINERAL	0	124					0.00
117	CEMENTO BLANCO TIPO HUASCARAN 49KG	BLANQUEADOR	0	52					0.00
118	PIEDRA GRANITO #2	MATERIALES	50	398					0.00
119	PIEDRA GRANITO #3	MATERIALES	50	438					0.00
12	PORCELANA PALO ROSA 2K	EMPORE	60	520					0.00
121	PIEDRA GRANITO #4	MATERIALES	0	20					0.00
122	CEMENTO HOLCIM 50KG	MATERIALES	960	0					6,928.74
123	CEMENTO BLANCO ALBORG 50KG.	MATERIALES	0	10					0.00
124	RESINA REDISPERSABLE RR-52100 (25KG)	MATERIALES	500	0					1,710.00
126	CARBONATO DE CALCIO MALLA 400 (25KG.)	MATERIALES	9,250	26					2,570.14
127	AZUL FTALOCIANINA 153 (25KG.)	MATERIALES	380	75					3,502.03
128	AMARILLO CROMO AL-73 (25KG.)	MATERIALES	870	0					3,380.70
129	OXIDO DE HIERRO ROJO R-4530 (25KG.)	MATERIALES	1,370	0					2,631.61
13	PORCELANA BEIGE 1K	EMPORE	0	2,601					0.00
130	OXIDO DE HIERRO AMARILLO Y-4021 (25KG.)	MATERIALES	1,620	0					4,243.75
131	OXIDO DE HIERRO NEGRO NB-4970 (25KG.)	MATERIALES	1,050	65					1,379.12
132	AZUL ULTRAMAR U-301 (25KG.)	MATERIALES	220	0					1,272.50
133	VERDE FTALOCIANINA 7 (25KG.)	MATERIALES	140	0					1,607.30
134	DIOXIDO DE TITANIO R-730 (25KG.)	MATERIALES	100	0					293.00
135	BERMOCOLL ME 1000X (20KG.)	MATERIALES	240	0					2,035.20
137	RESINA REDISPERSABLE 503 (25KG.)	MATERIALES	20,050	0					35,302.04
138	NEGRO DE HUMO POWCARBON 2849F (10KG.)	MATERIALES	560	0					2,648.70
139	OXIDO HIERRO AMARILLO 313 (25KG.)	MATERIALES	120	11					249.47
14	PORCELANA BEIGE 2K	EMPORE	460	5,250					0.00
140	OXIDO HIERRO ROJO 130R (25KG.)	MATERIALES	470	0					895.35
141	CIPEQ / ROJO SUDAPERM 2967C (20KG.)	MATERIALES	140	0					2,739.80
142	BERMOCOLL EBM 8000 (20KG.)	MATERIALES	40	0					380.00
147	ARENA FINA BAQUERO (50KG.)	MATERIALES	11,650	0					18,646.40
149	CEMENTO BLANCO AALBORG (50KG.)	MATERIALES	440	11					8,406.05
15	PORCELANA GRIS CLARO 1K	EMPORE	50,330	24,860					10,000.00
156	CARBONATO CALCIO 825C (50KG.) PARA EMPASTE	MATERIALES	1,030	0					3,235.05
157	EXPLOSA/ CARBONATO CRUZ AZUL MALLA 40 (50KG.)	MATERIALES	177,183	0					113,933.96
158	CALCAREOS / CARBONATO CALCIO MALLA 200 B SIN LAMINAR (50KG.)	MATERIALES	13,440	186					11,377.98
159	CALCAREOS / CARBONATO CALCIO MALLA 40 SIN LAMINAR (50KG.)	MATERIALES	48,190	0					16,692.24
16	PORCELANA GRIS CLARO 2K	EMPORE	61,821	37,621					16,354.10
161	CEGALATEX AD030C TANQUE 200KG.	MATERIALES	4,600	0					6,830.00
163	ADITIVO RHOLOGICO CANECA 20KG.	MATERIALES	40	0					175.50
164	ADITIVO BIOCIDO ACTICIDE SPX / NIPACIDE CF XS/ MERGAL CANECA 20KG.	MATERIALES	10	0					74.10
166	CEMENTO TIPO GUICHIMBORAZO (50KG.)	MATERIALES	58,420	48					79,261.76
167	SACO CEMENTO HOLCIM FUERTE TIPO GU (50KG.)	MATERIALES	172,170	910					222,044.53
17	PORCELANA DURAZNO 1K	EMPORE	0	70					0.00
170	FUNDA NATURAL 18X33X0.0007 GUARDAR PEGANTE	MATERIALES	20,210	0					8,971.36
171	SACO LAMINADO 40X62 ESTANDARS (20KG)	MATERIALES	87,060	0					18,284.49
172	SACO LAMINADO 40X62 PLUS (20KG.)	MATERIALES	13,830	0					2,904.30
173	SACO LAMINADO 40X62 PREMIUM (20KG.)	MATERIALES	37,720	0					7,922.88
174	SACO TRANSPARENTE LAMINADO 60X90 PORCELANA ESPAÑOLA	MATERIALES	5,000	0					1,661.00
177	CIREL (20KG.)	MATERIALES	0	0					16.00
178	FUNDAS PORCELANA ESPAÑOLA 9X12X0.004 (2KG.)	MATERIALES	28,150	0					12,536.11
18	PORCELANA DURAZNO 2K	EMPORE	0	282					0.00
180	FUNDAS PORCELANA ESPAÑOLA 8 1X4X10 1/4X0.0004 DE (1KG)	MATERIALES	6,070	0					3,896.44
183	FUNDAS PARA GUARDAR PEGANTE 18X33X0.0007	MATERIALES	130	0					5,169.31
184	ARENA DE RIO PARA PEGANTE (50KG.)	MATERIALES	12,840	0					16,084.20
186	SACO ESPANOL PREMIUM 45X60 (25KG.)	MATERIALES	6,900	121					1,493.36
189	REDISPERSABLE 4RK1460 DE (25KG.)	MATERIALES	20	0					78.75
19	PORCELANA MARFIL 1K	EMPORE	40	1,011					0.00
190	REDISPERSABLE DRK 550 DE (25KG.)	MATERIALES	20	0					86.25
191	IMAGOR / CELULOSICO DRK 2711 DE (25KG.)	MATERIALES	1,870	0					9,992.50
192	IMAGOR / CELULOSICOHPMC-MORTERO (25KG.)	MATERIALES	1,000	0					4,107.00
194	FLETE TRANSPORTE CARGA RIOBAMBA	SERVICIOS	60	1,260					0.00
196	CANECA RESINA INT.	MATERIALES	0	38					0.00
197	SALDO ANTERIOR	SERVICIOS	0	0					200.00
198	DOMINGUEZ PIEDRA RIO # 2	MATERIALES	230	0					308.10
20	PORCELANA MARFIL 2K	EMPORE	220	2,930					0.00
200	DOMINGUEZ PIEDRA RIO #3	MATERIALES	270	0					360.10
201	GREGORIO CARBONATO BLANQUEADOR 50KG.	MATERIALES	290	0					1,406.40
203	MINERVA / OXIDO DE HIERRO AMARILLO Y-2023 (50KGS.)	MATERIALES	570	0					1,495.00
204	CIPEQ / ROJO SUDACOLOR 570 (15KILOS) SACO	MATERIALES	150	0					2,760.00
205	FLETE TRANSPORTE CARGA SACOS PORCELANA QUEVEDO	MATERIALES	0	10					0.00

206	FLETE TRANSPORTE CARGA SACOS PEGANTE ESPAÑOL 20K QUEVEDO	MATERIALES	200	1,565					0.00
207	ENVASE 2KG BLANCO RESINA INTERIOR (80GRS)	MATERIALES	1,100						527.04
21	PORCELANA ROSE 1K	EMPORE	6	60					0.00
210	PEGANTE CONMACO PREMIUM 20KG	MATERIALES	6	416					0.00
211	PEGANTE CONMACO PLUS 20K	PEGANTE	6	300					0.00
212	FLETE TRANSPORTE CARGA SACOS PEGANTE ESPAÑOL 25KG TRONCAL	SERVICIOS	300	4,100					0.00
213	FLETE TRANSPORTE CARGA FDS. 2K PORCELANA ESPAÑOLA (TRONCAL)	EMPORE	2	937					0.00

214	FLETE TRANSPORTE CARGA EMPASTE SIKEPEGA 20K (TRONCAL)	EMPASTE	0	10					0.00
215	PORCELANA AZUL CIELO PARA PISCINA 2KG	EMPORE	0	65					0.00
216	FLETE TRANSPORTE PEGANTE ESPAÑOL 20K MANGLAR ALTO (300 SACOS)	SERVICIOS	0	150					0.00
219	FLETE CARGA TRANSPORTE PEGANTE ESPAÑOL 20K (150 SACOS)	SERVICIOS	0	3					0.00
22	PORCELANA ROSE 2K	EMPORE	0	60					0.00
220	PINTA FACIL 20K CON RESINA	BLANQUEADOR	0	6					0.00
224	FLETE TRANSPORTE PEGANTE ESPAÑOL QUINSALOMA 300 SACOS (20KG)	SERVICIOS	0	689					0.00
226	PORCELANA ROJO 2KG	EMPORE	0	30					0.00
227	FLETE TRANSPORTE CARGA PEGANTE PATRICIA PILAR (200 SACOS)	SERVICIOS	0	200					0.00
228	CIPEQ / ROJO SUDAFast 331-15KGS	MATERIALES	15	0					288.15
229	ESPESANTE DE MORTERO	MATERIALES	1,250	5					6,902.28
230	BOBINA BLANCA BKG NE 12 4 S	MATERIALES	22	0					1,060.39
231	CHUMACERA UCF211-32 (2") H. FUNDIDO	MATERIALES	2	0					52.38
232	PIÑON	MATERIALES	1	0					58.13
233	CIREL PREMIUM 25KG	MATERIALES	1	0					10.00
234	CIREL STANDARD 25KG	MATERIALES	1	0					10.00
235	MAPRECU CEMENTO BLANCO ARGO	MATERIALES	4,000	2					1,686.16
236	MOTOR TRIFASICO SHIP CERRADO 2 POLOS	MATERIALES	1	0					413.62
237	OXIDO HIERRO ROJO (CAOBA)	POLVO MINERAL	0	199					0.00
238	AMARILLO CROMO MEDIO	POLVO MINERAL	0	12					0.00
239	CAFE NOGAL	POLVO MINERAL	0	11					0.00
240	TIZA ULTRA BLANCA FINA	MATERIALES	0	11					0.00
241	TIZA PROFESIONAL CREMOSA	MATERIALES		5					0.00
242	SACO 6X100 LAMINADO BLS	MATERIALES	100	0					32.00
243	NUTRICALCIO 200 45KG	MATERIALES	100	0					300.00
245	POLVO MINERAL TIÑEMAX CELESTE 25KG.	POLVO MINERAL	2	8					0.00
247	CELULOSICO HPHC-PINTURAS	MATERIALES	300	0					1,192.50
248	MULTO DE SACOS VACIOS	MATERIALES		82					0.00
249	ESPESANTE DE MORTERO PREMIUM	MATERIALES	1,225	0					7,346.20
25	PORCELANA SALMON 1K	EMPORE	0	2,506					0.00
250	CEMENTO BLANCO ARGO	MATERIALES	180	77					3,721.71
251	BERMOCOLL	MATERIALES	0	11					0.00
252	MOSTRADOR EN MADERA LAMINADA	MATERIALES	100	0					804.00
253	OXIDO HIERRO CAFE 663 (HYROX) - 25 KGS	MATERIALES	50	0					81.50
254	OXIDO HIERRO NEGRO	MATERIALES		17					0.00
255	CHUMACERA TRANSLINK UCF209-28 1.3/4	MATERIALES	1	0					24.47
256	CHUMACERA TRANSLINK UCP213-40 2.1/2	MATERIALES	1	0					48.40
257	CIPEQ MELLUTEL EM200MS - 25 KGS	MATERIALES	7,896	0					42,293.73
258	CIREL 174606 - ESPAÑOL STANDARD CERAMICA 25KG - VERDE- BASE	MATERIALES	1	0					254.83
259	CIREL 171649 - ESPAÑOL PREMIUM PORCELANATO 25KG - ROJO- BASE	MATERIALES	1	0					99.88
26	PORCELANA SALMON 2K	EMPORE	6,375	3,684					2,372.18
260	CIREL 174603 - ESPAÑOL PLUS CERAMICA SOBRE CERAMICA 25 KG AZUL-BASE	MATERIALES	1	0					151.36
261	CEMENTO BLANCO ARGO 40K	MATERIALES	1,450	90					19,633.02
262	40XS6 F/PLANO BLANCO LAM BRILL "ESPAÑOL STANDARD 20KG"	MATERIALES	117,452	0					34,401.69
264	CARBONATO DE CALCIO MALLA 100 POLVO	MATERIALES	10,265	0					16,231.05
265	NG NEGRO ESENCIA	POLVO MINERAL	0	50					0.00
266	40XS6 F/PLANO BLANCO LAM BRILL "ESPAÑOL PREMIUM 20KG"	MATERIALES	66,052	0					19,346.64
267	40XS6 F/PLANO BLANCO LAM BRILL "ESPAÑOL PLUS 20KG"	MATERIALES	40,780	0					7,463.09
268	ANTIESPUMANTE (DAPRO DF 7010)	MATERIALES	1	0					33.56
269	MONOETINEL GLICOL	MATERIALES	1	0					39.00
27	PORCELANA SAHARA 1K	EMPORE	0	1,319					0.00
270	FUNDAS 20X 30 NATURAL	MATERIALES	200	0					16.00
271	POLVO MINERAL NEGRO ESENCIA	POLVO MINERAL	0	16					0.00
272	EXPLOSA/ M 40 SACOS BLANCOS CRUZ AZUL	MATERIALES	60	0					105.00
273	EMPASTE SIKEPEGA EXTERIOR 20K	EMPASTE	16	223					0.00
274	LIBRA AMARILLO ESENCIA	POLVO MINERAL	0	25					0.00
275	OXIDO AMARILLO OCRE	POLVO MINERAL	0	24					0.00
276	EMPASTE INTERIOR SIN RESINA	EMPASTE	0	1					0.00
277	PASADA DE NEGRO 50K	POLVO MINERAL	15	0					75.00
278	PASADA DE ROJO FUERTE 50K	POLVO MINERAL	6	0					30.00
279	PASADA DE ROJO SUAVE	POLVO MINERAL	6	0					30.00
28	PORCELANA SAHARA 2K	EMPORE	6,105	3,226					893.64
280	PASADA DE AZUL FUERTE 50K	POLVO MINERAL	6	0					30.00
281	PASADAS DE AZUL SUAVE 50K	POLVO MINERAL	3	0					15.00
282	GREGORIO BLANQUEADO NATURAL 50K	BLANQUEADOR	36	0					172.80
283	OXIDO HIERRO NARANJA 960 (HYROX) - 25KGS	MATERIALES	75						134.25
285	8.25 X 10.25 X 0.004 FDAS P.ESPAÑOLA 1KG - SBX 3.000 C/U + 1BX 2	MATERIALES	58						2,674.92
286	CIPEQ / AZUL FTALOCIANINA M3L4 15:3 (MALLAK) - 25 KGS	MATERIALES	25						264.50
287	CIPEQ / VERDE FTALOCIANINA M7L6 (MALLAK) - 25 KGS	MATERIALES	25	43					241.25
288	DIOXIDO DE TITANIO AOTIOX R04 V230040431	MATERIALES	75	5					202.50
289	CIREL 184452 - ESPAÑOL STANDARD CERAMICA 25 KG - VERDE - BASE	MATERIALES	1						10.00
29	PORCELANA VAINILLA 1K	EMPORE	0	342					0.00
290	CEGALATEX VA055PLUS EN TANQUE DE 200 KG	MATERIALES	600						1,350.00
291	CANECA C/P 5 GAL	MATERIALES	0						0.00
292	CIPEQ / ETILENGLICOL	MATERIALES	40						57.60
293	40XS6 F/PLANO BLANCO LAM VRILL "ESPANOL STANDARD 25KG"	MATERIALES	23,683						6,936.75
294	SEGUPLAST / 9 X 12.75 X 0.004 FDAS P.ESPAÑOLA 2KG - 48X 2.000 C/U	MATERIALES	55						3,659.97
295	CIPEQ / EC-CELL HPS 150 - 25 KGS	MATERIALES	225						937.50
296	MINERVA / VERDE FTALOCIANINA 0761C	MATERIALES	20						236.00
297	DON RAFA / SACO BLANCO 10 KLS	MATERIALES	100						22.00
298	DON RAFA / SACO BLANCO 20 KLS	MATERIALES	100						28.00
299	DON RAFA / FUNDAS PLASTICAS 20 KLS	MATERIALES	1,000						30.00
30	PORCELANA VAINILLA 2K	EMPORE	36	799					0.00
300	40XS6 F/PLANO BLANCO LAM BRILL "ESPAÑOL PREMIUM 25KG	MATERIALES	13,021						3,613.85
301	SACO DE BLANQUEADOR 50 KILOS	MATERIALES	81						388.80
302	ESPESANTE ACRILICO	MATERIALES	60						240.00
303	MONOETILENGLICOL	MATERIALES	40						66.00
304	SACO BLANCO LAMINADO 60 X 100	MATERIALES	300						90.00
305	NEGRO DE HUMO BP280	MATERIALES	25						148.75
306	NEGRO DE HUMO BEBLACK-20KG	MATERIALES	20						135.60
307	DIOXIDO DE TITANIO R900	MATERIALES	75						232.50
308	DIOXIDO DE TITANIO R-2295 - 25KG	MATERIALES	25						72.25
309	CEMENTO BLANCO HUASCARAN SACO 25 KG	MATERIALES	105	2					1,435.99
31	PORCELANA MELON 1K	EMPORE	0	130					0.00
310	PASADA DE COLORES	MATERIALES	15						75.00
311	POLVO MINERAL TIÑEMAX ROSADO 25KG	POLVO MINERAL		1					0.00
312	AZUL FTALOCIANINA 1534C (20)	MATERIALES	20						20.00
313	FIBRA DE CELULOSA 10W-500 - 15KGS	MATERIALES	75						311.25
314	SUPERPLASTIFICANTE PCB030-25KGS	MATERIALES	75						387.40
315	CALCAREOS CARBONATO DE CALCIO MALLA 30 PRESENTACION SACOS DE 50 KG SIN LAMINAR	MATERIALES	2,757						4,549.05
316	CIPEQ / RESINA REDISPERSABLE 760 F-25KGS	MATERIALES	25						67.00
317	CIPEQ / SILANE AGENT S81 - NDCHEM- 25KGS	MATERIALES	3						30.78
32	PORCELANA MELON 2K	EMPORE	0	219					0.00
33	PORCELANA CARAMELO 1K	EMPORE	180	7,852					0.00
34	PORCELANA CARAMELO 2K	EMPORE	9,495	13,431					2,652.10
35	PORCELANA BLANCO HUESO 1K	EMPORE	0	150					0.00
36	PORCELANA BLANCO HUESO 2K	EMPORE	15	1,109					0.00
37	PORCELANA NARANJA 1K	EMPORE	0	370					0.00
38	PORCELANA NARANJA 2K	EMPORE	51	1,191					0.00
39	PORCELANA FRESA 1K	EMPORE	0	95					0.00

4	PORCELANA BLANCO 2K	EMPORE	58,912	27,628				32,862.80
40	PORCELANA FRESA 2K	EMPORE	0	15				0.00

16.2 Anexo 2 Script en Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, silhouette_samples, davies_bouldin_score

np.random.seed(42)
MAROON, BLUE, GOLD, GREEN = '#8B1E3F', '#1B3A6B', '#C9A227', '#4C9A6A'
PALETTE = [MAROON, BLUE, GOLD, GREEN]
```

```
#cargamos datos desde la pc
from google.colab import files
uploaded = files.upload()
```

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 Super_Kardex_General_2023-2025.xlsx.xlsx to Super_Kardex_General_2023-2025.xlsx.xlsx

```
df = pd.read_excel('Super_Kardex_General_2023-2025.xlsx.xlsx', sheet_name='Super_Kardex_General', header=2)
df = df.rename(columns={'2023-12-31 00:00:00.000': 'fecha', 'CEMENTO BLANCO HUASCARAN (50KG.)': 'producto'})
df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
```

```
cols_check = [c for c in df.columns if c not in ['Unnamed: 9']]
filas_antes = df.dropna(subset=['producto']).shape[0]
df = df.drop_duplicates(subset=cols_check).copy()
filas_despues = df.dropna(subset=['producto']).shape[0]
```

```
print(f"Filas con producto antes de deduplicar : {filas_antes}")
print(f"Filas con producto después de deduplicar: {filas_despues}")
```

```
Filas con producto antes de deduplicar : 26956
Filas con producto después de deduplicar: 17863
```

```
# Construcción de variables (demanda completa, costo en cascada)
tx = df.dropna(subset=['producto', 'fecha']).copy()
tx['mes'] = tx['fecha'].dt.to_period('M')
tx['SALDO ANTERIOR'] = pd.to_numeric(tx['SALDO ANTERIOR'], errors='coerce')
tx[28.6071] = pd.to_numeric(tx[28.6071], errors='coerce')
tx['28.6071.1'] = pd.to_numeric(tx['28.6071.1'], errors='coerce')
tx[1859.464] = pd.to_numeric(tx[1859.464], errors='coerce')

master = df.dropna(subset=['producto']).groupby('producto').agg(
    Articulo=('producto', 'first'), Linea=('MATERIALES', 'first')).reset_index()

# Demanda / frecuencia / CV / intermitencia: histórico COMPLETO deduplicado
mensual = tx.groupby(['producto', 'mes'])[0.148].sum().reset_index()
all_months = pd.period_range(tx['mes'].min(), tx['mes'].max(), freq='M')
piv = mensual.pivot(index='producto', columns='mes', values=0.148).reindex(columns=all_months, fill_value=0).fillna(0)
```

```
demanda_total = piv.sum(axis=1)
media_mensual = piv.mean(axis=1)
std_mensual = piv.std(axis=1)
cv_demanda = (std_mensual / media_mensual.replace(0, np.nan)).fillna(0)
intermitencia = (piv == 0).mean(axis=1)
frecuencia = tx[tx[0.148]>0].groupby('producto').size().reindex(demanda_total.index, fill_value=0)
```

```
# Saldo / rotación: histórico completo (saldo negativo = señal real, no se filtra)
saldo_prom = tx.groupby('producto')['SALDO ANTERIOR'].mean()
```

```

# Saldo / rotación: histórico completo (saldo negativo = señal real, no se filtra)
saldo_prom = tx.groupby('producto')['SALDO ANTERIOR'].mean()
pct_saldo_neg = tx.groupby('producto').apply(lambda g: (g['SALDO ANTERIOR']<0).mean(), include_groups=False)
rotacion = (demanda_total / saldo_prom.replace(0, np.nan)).fillna(0)

# Costo unitario: cascada de validez (Promedio -> Ultimo -> Precio), sin borrar filas
costo_prom = tx[tx[28.6071]>0].groupby('producto')[28.6071].mean()
costo_ult = tx[tx[28.6071.1]>0].groupby('producto')[28.6071.1].mean()
precio = tx[tx[1859.464]>0].groupby('producto')[1859.464].mean()
pct_costo_valido = tx.groupby('producto').apply(lambda g: (g[28.6071]>0).mean(), include_groups=False)

costo_final = costo_prom.reindex(demanda_total.index).fillna(costo_ult).fillna(precio)
valor_consumido = demanda_total * costo_final

feat = pd.DataFrame({
    'demanda_total': demanda_total, 'frecuencia_salidas': frecuencia,
    'cv_demanda': cv_demanda, 'intermitencia': intermitencia,
    'costo_unit_prom': costo_final, 'valor_consumido': valor_consumido,
    'saldo_promedio': saldo_prom, 'rotacion_inventario': rotacion,
    'pct_costo_valido': pct_costo_valido, 'pct_saldo_negativo': pct_saldo_neg,
}).reset_index().merge(master, on='producto', how='left')

feat = feat.replace([np.inf, -np.inf], np.nan).dropna(subset=['costo_unit_prom']).reset_index(drop=True)

# Transformación, split train/test y estandarización
vars8 = ['demanda_total', 'frecuencia_salidas', 'cv_demanda', 'intermitencia',
        'costo_unit_prom', 'valor_consumido', 'saldo_promedio', 'rotacion_inventario']
log_vars = ['demanda_total', 'frecuencia_salidas', 'costo_unit_prom', 'valor_consumido', 'saldo_promedio']

X = feat[vars8].copy()
for v in log_vars:
    X[v] = np.log1p(X[v].clip(lower=0))
X['rotacion_inventario'] = np.arcsinh(X['rotacion_inventario']) # puede ser negativo

train_idx, test_idx = train_test_split(feat.index, test_size=0.2, random_state=42)
X_train, X_test = X.loc[train_idx], X.loc[test_idx]
feat_train, feat_test = feat.loc[train_idx].copy(), feat.loc[test_idx].copy()

# Imputar NaN con la mediana de X_train o dropear si toda la columna es NaN
cols_to_drop = []
for col in X_train.columns:
    if X_train[col].isnull().any():
        median_val = X_train[col].median()
        if pd.isna(median_val): # Si la mediana es NaN, la columna entera es NaN en X_train
            cols_to_drop.append(col)
    else:
        X_train[col] = X_train[col].fillna(median_val)
        X_test[col] = X_test[col].fillna(median_val)

# Dropear las columnas identificadas como completamente NaN
if cols_to_drop:
    print(f"Columnas completamente NaN en X_train y serán eliminadas: {cols_to_drop}")
    X_train = X_train.drop(columns=cols_to_drop)
    X_test = X_test.drop(columns=cols_to_drop)

scaler = StandardScaler().fit(X_train) # fit SOLO en train
Xz_train, Xz_test = scaler.transform(X_train), scaler.transform(X_test)

... Columnas completamente NaN en X_train y serán eliminadas: ['saldo_promedio']

```



```

#ACP
pca = PCA(random_state=42).fit(Xz_train)
evr, eigvals = pca.explained_variance_ratio_, pca.explained_variance_

n_comp = max(int((eigvals>1).sum()), 2) # criterio de Kaiser
loadings = pd.DataFrame(pca.components_[0:n_comp].T, index=X_train.columns,
                        columns=[f'CP{i+1}' for i in range(n_comp)])

scores_train_full = pca.transform(Xz_train)
scores_test_full = pca.transform(Xz_test)
scores_train, scores_test = scores_train_full[:,0:n_comp], scores_test_full[:,0:n_comp]

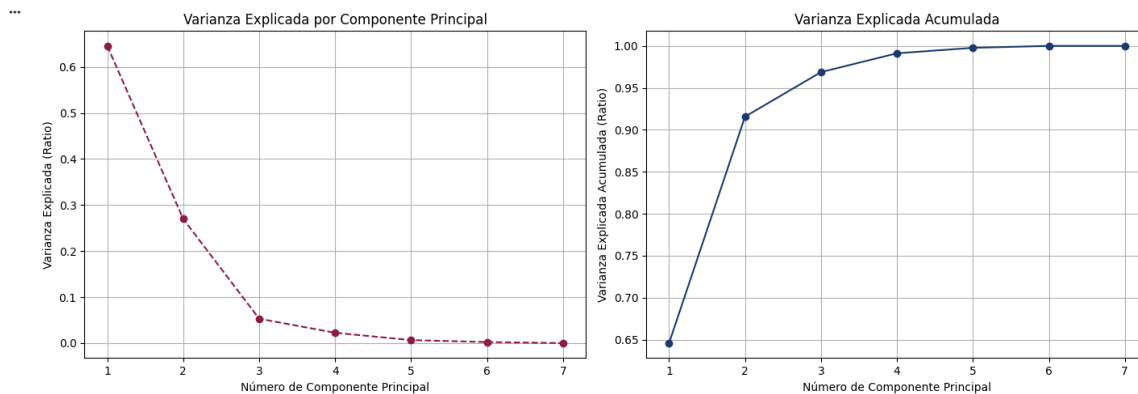
# Gráfico de la varianza explicada por cada componente principal
fig, ax = plt.subplots(1, 2, figsize=(14, 5))

ax[0].plot(range(1, len(evr) + 1), evr, marker='o', linestyle='--', color=MAROON)
ax[0].set_title('Varianza Explicada por Componente Principal')
ax[0].set_xlabel('Número de Componente Principal')
ax[0].set_ylabel('Varianza Explicada (Ratio)')
ax[0].grid(True)

ax[1].plot(range(1, len(evr) + 1), np.cumsum(evr), marker='o', linestyle='-', color=BLUE)
ax[1].set_title('Varianza Explicada Acumulada')
ax[1].set_xlabel('Número de Componente Principal')
ax[1].set_ylabel('Varianza Explicada Acumulada (Ratio)')
ax[1].grid(True)

plt.tight_layout()
plt.show()

```



```

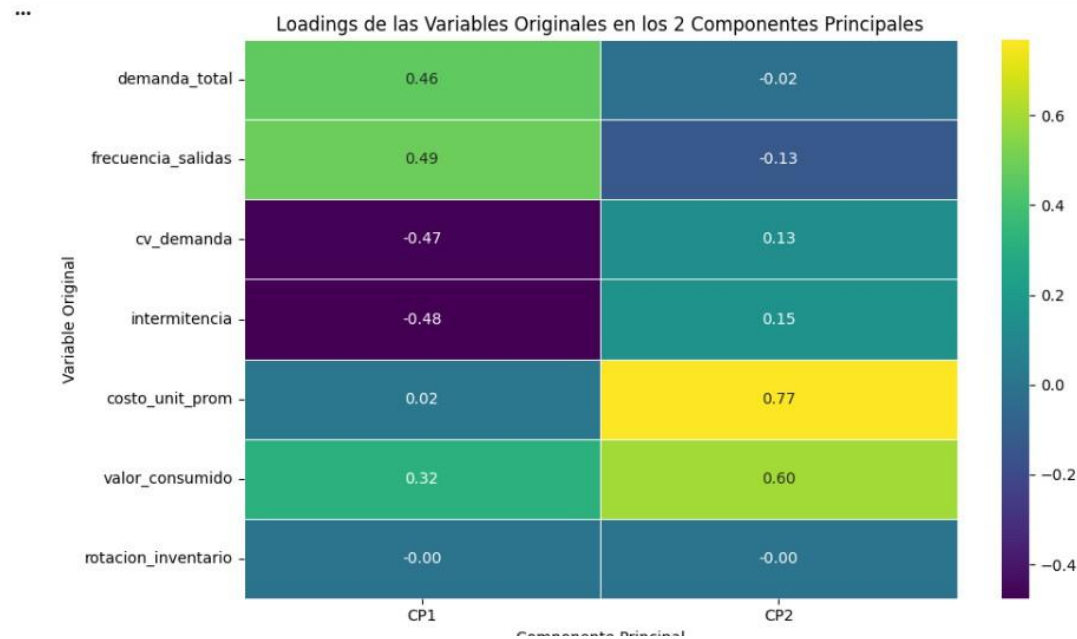
# Gráfico de los Loadings para los componentes principales seleccionados
fig = plt.figure(figsize=(10, 6))
sns.heatmap(loadings, cmap='viridis', annot=True, fmt=".2f", linewidths=.5)
plt.title(f'Loadings de las Variables Originales en los {n_comp} Componentes Principales')
plt.xlabel('Componente Principal')
plt.ylabel('Variable Original')
plt.tight_layout()

```

```

# Gráfico de los Loadings para los componentes principales seleccionados
fig = plt.figure(figsize=(10, 6))
sns.heatmap(loadings, cmap='viridis', annot=True, fmt=".2f", linewidths=.5)
plt.title(f'Loadings de las Variables Originales en los {n_comp} Componentes Principales')
plt.xlabel('Componente Principal')
plt.ylabel('Variable Original')
plt.tight_layout()
plt.show()

```



```

# @title step_artifacts
num_fig = "2" # @param {type:"string"}
step = 'ModelOptimization' # @param ["DataLoading", "DataExploration", "DataCleaning", "DataWrangling", "DataVisual:
# upload_plt_to_gcs(num_fig, step, fig) # This line was commented out because the function is not defined

#Selección de k (codo + silueta) y ajuste final
ks = range(2,9)
inertias, sils = [], []
for k in ks:
    kmk = KMeans(n_clusters=k, random_state=42, n_init=20).fit(scores_train)
    inertias.append(kmk.inertia_)
    sils.append(silhouette_score(scores_train, kmk.labels_))

K_ELEGIDO = 4
kmeans_final = KMeans(n_clusters=K_ELEGIDO, random_state=42, n_init=20).fit(scores_train)
labels_train = kmeans_final.labels_
labels_test = kmeans_final.predict(scores_test) # asignación por centroide más cercano

```

n


```

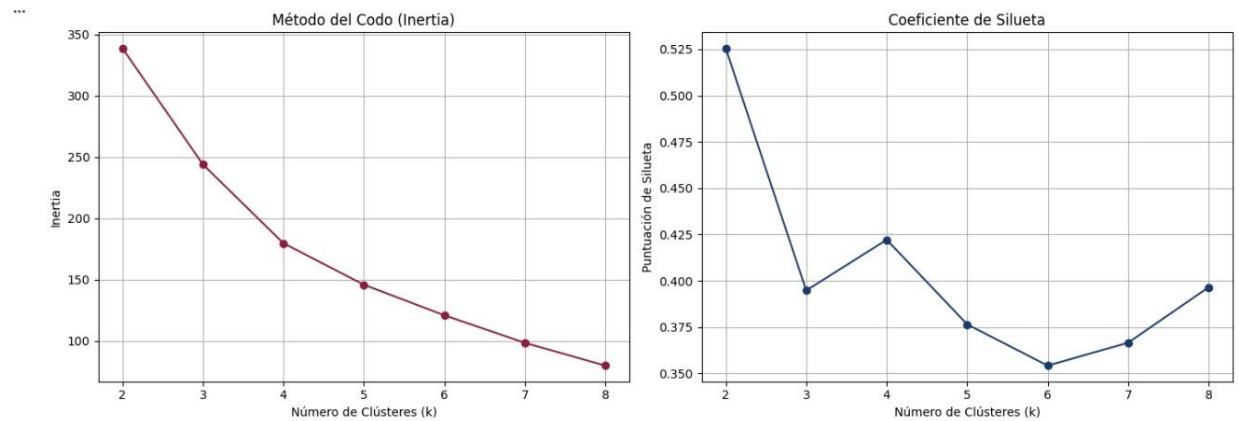
# Gráfico del método del codo y coeficiente de silueta
fig, ax = plt.subplots(1, 2, figsize=(14, 5))

ax[0].plot(ks, inertias, marker='o', color=MAROON)
ax[0].set_title('Método del Codo (Inertia)')
ax[0].set_xlabel('Número de Clústeres (k)')
ax[0].set_ylabel('Inertia')
ax[0].grid(True)

ax[1].plot(ks, sils, marker='o', color=BLUE)
ax[1].set_title('Coeficiente de Silueta')
ax[1].set_xlabel('Número de Clústeres (k)')
ax[1].set_ylabel('Puntuación de Silueta')
ax[1].grid(True)

plt.tight_layout()
plt.show()

```



```

# @title step_artifacts
num_fig = "3" # @param {type:"string"}
step = 'ModelEvaluation' # @param ["DataLoading", "DataExploration", "DataCleaning", "DataWrangling", "DataVisualization"]
# upload_plt_to_gcs(num_fig, step, fig) # This line was commented out because the function is not defined

#Evaluación train vs. test
sil_train = silhouette_score(scores_train, labels_train)
sil_test = silhouette_score(scores_test, labels_test)
db_train = davies_bouldin_score(scores_train, labels_train)
db_test = davies_bouldin_score(scores_test, labels_test)

#Perfil de clústeres
feat_train['cluster'], feat_train['split'] = labels_train, 'train'
feat_test['cluster'], feat_test['split'] = labels_test, 'test'
resultado = pd.concat([feat_train, feat_test], ignore_index=True)

perfil = resultado.groupby('cluster')[vars8].mean()
perfil['n_productos'] = resultado.groupby('cluster').size()

```

```

# Mostrar el perfil de clústeres
print("Perfil de Clústeres (valores promedio de las variables originales y número de productos por clúster):")
print(perfil)

```

```

... Perfil de Clústeres (valores promedio de las variables originales y número de productos por clúster):
      demanda_total  frecuencia_salidas  cv_demanda  intermitencia \
cluster
0      1539.619048      5.976190      2.788536      0.835238
1       859.112000      3.349398      3.301784      0.889639
2     26368.000000     336.500000      0.375584      0.000000
3     21211.785714     300.404762      0.672913      0.157143

      costo_unit_prom  valor_consumido  saldo_promedio \
cluster
0       49.135890      4.729461e+04      NaN
1       3.315500      1.884048e+03      NaN
2     9452.000000      3.132723e+08      NaN
3       3.013286      5.478701e+04      NaN

      rotacion_inventario  n_productos
cluster
0              0.0              42
1              0.0              83
2              0.0               2
3              0.0              42

```

```

# Visualización de los clústeres en el espacio PCA
plt.figure(figsize=(10, 8))

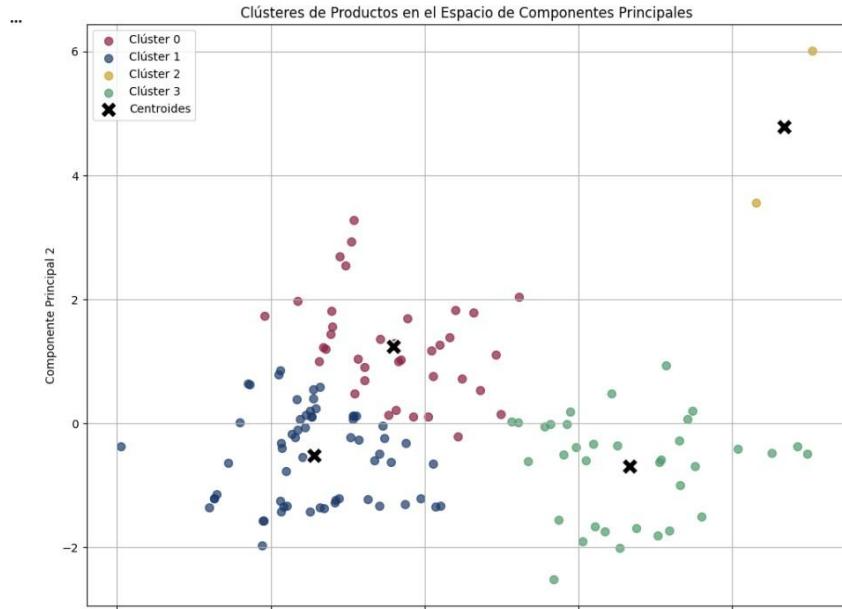
# Obtener los centroides de los clústeres en el espacio PCA
# Se necesitan los centroides originales del KMeans en el espacio de scores_train
centroids = kmeans_final.cluster_centers_

# Crear un scatter plot para cada clúster
for i in range(K_ELEGIDO):
    plt.scatter(scores_train[labels_train == i, 0],
               scores_train[labels_train == i, 1],
               label=f'Clúster {i}',
               alpha=0.7,
               s=50, # Tamaño de los puntos
               c=PALETTE[i % len(PALETTE)]) # Usar la paleta de colores

# Graficar los centroides de los clústeres
plt.scatter(centroids[:, 0], centroids[:, 1],
           marker='X', s=200,
           color='black', edgecolor='white',
           label='Centroides', zorder=2)

plt.title('Clústeres de Productos en el Espacio de Componentes Principales')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```



Article AutoML Approach to Stock Keeping Units Segmentation

Institute of Technology, 1 Amherst Street, Cambridge, MA 02142, USA; ilyajack@mit.edu

Ilya Jackson 

Abstract: A typical retailer carries 10,000 stock-keeping units (SKUs). However, these numbers may exceed hundreds of millions for giants such as Walmart and Amazon. Besides the volume, SKU data can also be high-dimensional, which means that SKUs can be segmented on the basis of various attributes. Given the data volumes and the multitude of potentially important dimensions to consider, it becomes computationally impossible to individually manage each SKU. Even though the application of clustering for SKU segmentation is common, previous studies do not address the problem of parametrization and model finetuning, which may be extremely tedious and timeconsuming in real-world applications. Our work closes the research gap by proposing a solution that leverages automated machine learning for the automated cluster analysis of SKUs. The proposed framework for automated SKU segmentation incorporates minibatch K-means clustering, principal component analysis, and grid search for parameter tuning. It operates on top of the Apache Parquet file format, an efficient, structured, compressed, column-oriented, and big-data-friendly format. The proposed solution was tested on the basis of a real-world dataset that contained data at the pallet level.

Keywords: SKU; clustering; AutoML

Introduction

Supply chains encompass manufacturing procurement, material handling, transportation, and distribution [1]. Demand propagates upstream as the information flow and supply responses in the form of commodity flow. However, supply cannot immediately match the demand due to various physical limitations involved in this process. For example, factories can produce only a certain number of items, trucks cannot exceed speed limits, and the capacity and number of container ships are constrained. The lead times might also be several weeks or even months in global supply chains. In this regard, inventory management is essential, since the inventory plays the role of a buffer against uncertainty and ensures business continuity across supply chains [2].

The importance of managing inventory is also illustrated by the US Department of Commerce, which regularly keeps track of inventory value held in different parts of the supply chain. In April 2022, USD 2.27 trillion were held in inventory across all types of business [3]. The retail industry alone accounted for USD 662 billion, approximately 17% of which was held by e-commerce [4]. As Figure 1 shows, these values recovered from the pandemic shocks and are on the rise.

Humanity has engaged in inventory management, at least in some primitive form, since it started gathering and stockpiling the planet's resources. However, the rigorous approach of prioritizing items for management attention can be dated back to Joseph Moses Juran [5]. Since a typical inventory system could contain an immense number of items with unique characteristics such as price, demand, and physical properties, analyzing them individually would outstrip anyone's managerial effort and attention. That is why retailers frequently use stock-keeping-unit (SKU) segmentation to guide themselves in planning, buying, and allocating products [6]. The common purposes for SKU segmentation include:

- Selecting an appropriate inventory control policy [7,8].



Citation: Jackson, I. AutoML Approach to Stock Keeping Units Segmentation. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 1512–1528. <https://doi.org/10.3390/jtaer17040076>

Academic Editor: Eduardo Álvarez-Miranda

Received: 11 September 2022

Accepted: 14 November 2022

Published: 15 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author.

Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Center for Transportation & Logistics, Massachusetts

- Understanding the demand patterns [\[9\]](#).

- Optimal allocation under the limited storage capacity [10].
- Selecting an appropriate supply-chain strategy [9].
- Evaluating the environmental impacts of inventory throughout its life cycle [11].

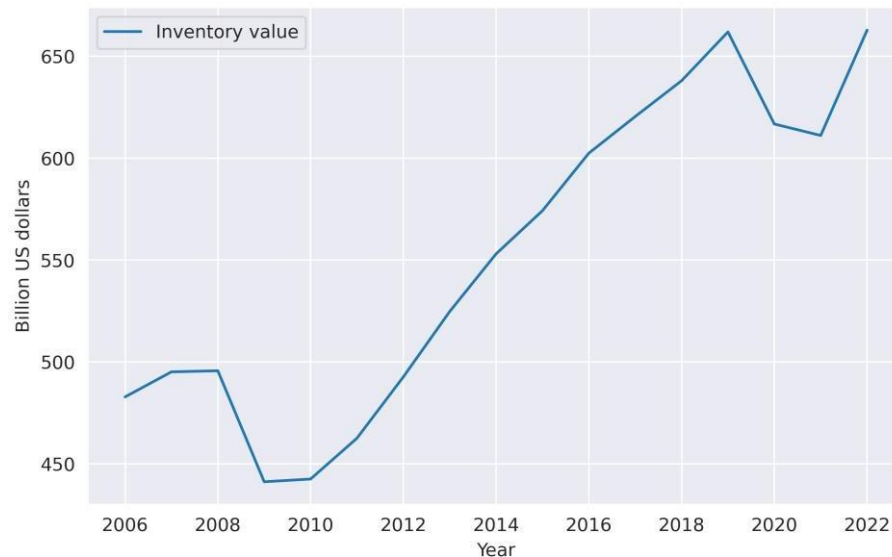


Figure 1. Value of inventories held for sale at the retail level. The figure was produced on the basis of data reported by the U.S. Census Bureau [3].

Every customer sees SKUs on a daily basis as barcodes in the grocery stores or items on the e-commerce platforms. A typical retailer carries 10,000 SKUs [6]. For giants such as Walmart, the number exceeds 75 million [12]. However, the e-commerce industry typically operates with an even more extensive catalog of items. A genuinely gargantuan case is the Amazon marketplace, acting as the home for more than 350 million SKUs [13]. Considering the extension of the supply chain directly into inhand devices through the adoption of digital technologies, and consumers' impatience, the future scale of the ecommerce industry can be expected to skyrocket.

Besides the volume, SKU data can also be distinguished by the high dimensionality, which means that SKUs can be segmented on the basis of various attributes. The most traditional way is the ABC classification method, which segments the SKUs on the basis of sales volume. The underlying theory of this segmentation method is the 20–80 Pareto principle. The principle suggests that 20% of SKUs generate approximately 80% of total revenue. Therefore this 20% should constitute the category that deserves the most attention from management. ABC segmentation utilizes only one dimension to segment the SKUs that, on the first hand, facilitates implementation [14]. However, on the other hand, SKUs within the segment can have substantially different characteristics. For example, two SKUs may have similar contributions to the revenue, but one may have a high unit price but low demand, and the other may have a low unit value but high demand. Indeed, it would be wrong to manage these SKUs under the same strategy. The physical properties of the SKU are another example. ABC segmentation does not consider perishability, which is essential for grocery retail. The weight and height of the loaded pallet can be crucial if the business operates under the constrained storage space, which is the case for retail chains in dense urban areas [10].

Given the data volumes and the multitude of potentially important dimensions to consider, it becomes computationally impossible to individually manage each

SKU. In this regard, the prominent question arises: “How to aggregate SKUs so that the resulting management policies are close enough to the case where each SKU is considered individually?” Our paper aims to answer this long-standing question by taking advantage of state-of-the-art big-data technologies and automated machine-learning (AutoML) approaches [15].

The remainder of this paper is organized as follows. Section 2 presents the related work and highlights the research gap. Section 3 introduces the methodology at a high level and provides the rationale behind the choice of the individual components. Section 4 demonstrates how the proposed framework functions on the basis of a real-world dataset. Section 5 highlights the potential advantages of the proposed framework, and discusses several use cases along with managerial implications. Lastly, Section 6 summarizes the paper and suggests promising directions for future research.

Related Work and Novelty

Considering the size and dimensionality of SKU data, and the market pressure on businesses to make critical decisions quickly, it is frequently not computationally feasible to treat every item individually. An alternative would be to define groups while taking into account all product features that notably influence the operational, tactical, or strategic problem of interest. These features can vary depending on the problem of interest, and go beyond the revenue and demand attributes used in classical ABC analysis. Clustering in particular and unsupervised machine learning in general are the most natural approaches to creating groups of SKUs that share similar characteristics [16].

Early attempts to group SKUs using cluster analysis go back to Ernst and Cohen, who proposed the operations related groups method to consider a full range of operationally significant item attributes. The method is similar to factor analysis and allows for considering a full range of operationally significant item attributes. The method was initially designed to assist an automobile manufacturer with developing inventory control policies [7]. Shortly after that, Srinivasan and Moon introduced a hierarchical-clustering-based methodology to facilitate inventory control policies in supply chains [17]. Hierarchical clustering still remains a popular technique. For instance, in a recent study, the authors proposed a hierarchical segmentation for spare-part inventory management [18]. Even though such an approach is simple to implement and interpret, hierarchical clustering has several fundamental shortcomings. First, the method does not scale well because of the computational complexity compared to more efficient algorithms, such as K-means. Second, hierarchical clustering is biased towards large clusters and tends to break them. Third, in hierarchical clustering, the order of the data impacts the final results. Lastly, the method is sensitive to noise and outliers [19].

Currently, K-means is the most popular clustering technique [20]. The algorithm is appealing in many aspects, including computational and memory efficiency, and interpretability in low-dimensional cases. The notable applications to SKU segmentation include Cannata et al., who combined K-means with self-organizing maps to cluster products across multiple dimensions [21]. Wu et al. applied K-means along with correlation analysis to identify primary indicators to predict demand patterns [22]. Egas and Masel used K-means clustering to reduce the time required to derive the inventory-control parameters in a largescale inventory system. The proposed method outperformed a demand-based assignment strategy [23]. Ozturk et al. used K-means to obtain SKU clusters. Then, an artificial neural network was applied to forecast demand for each individual cluster [24].

It is also essential to highlight the applications of fuzzy C-means to SKU segmentation. Fuzzy C-means is a form of clustering in which a cluster is represented by a fuzzy set. Therefore, each data point can belong to more than one cluster. Notable applications include multicriteria ABC analysis using fuzzy C-means clustering [25]. A recent study by Kucukdeniz and Sonmez proposed the integrated order-picking strategy based on SKU segmentation through fuzzy C-means. In this study, fuzzy clustering was applied to consider the order frequency of SKUs and their weights while designing the warehouse layout [26]. Fuzzy C-means is an interesting alternative to K-means. However, the fact that each SKU can belong to more than one cluster is a disadvantage in the business context. Since businesses must quickly and confidently make critical yes/no decisions, the fuzziness of SKU segments does not provide any helpful flexibility for decision making at scale, and introduces unnecessary layers of complexity. Additionally, the fuzzy C-means method has the same drawbacks as those of classical K-means. The drawbacks include the a priori specification of the number of clusters, poor performance on datasets that contain clusters with unequal sizes or densities, and sensitivity to noise and outliers [27]. In order to overcome the aforementioned shortcomings, preprocessing and dimensionality reduction techniques like PCA are applied to SKU data prior to clustering. For example, the authors conducted a study on the application of a constrained clustering method reinforced with principal component analysis. According to the research, the proposed method can provide significant compactness among item clusters [28]. Another study took advantage of PCA and K-means to segment SKUs for recommendation purposes [29]. In recent work, Bandyopadhyay et al. used PCA prior to K-means to achieve the effective segmentation of both customers and SKUs in the context of the fashion e-commerce industry [29].

Table 1 lists the aforementioned clustering-based approaches to SKU segmentation and groups them in accordance with the utilized method. Even though the combination of K-means and PCA helps in overcoming some critical problems such as sensitivity to noise and multicollinearity, it does not address the parametrization problem, namely, the a priori specification of the number of clusters. Moreover, parametrization becomes even more sophisticated because the potential user must choose the number of principal components for PCA. The problem of parametrization and model finetuning can be extremely tedious and time-consuming in real-world applications. That is why our work attempts to close the research gap by proposing a framework that leverages AutoML for the automated cluster analysis of SKUs.

Table 1. Approaches to SKU segmentation.

Segmentation Method	Publication
Operations related groups	[7]
Hierarchical clustering	[17,18]
K-means	[21–24]
Fuzzy C-means	[25–27]
K-means with PCA	[28,29]

Materials and Methods

Summarizing the related work, we represent the data pipeline in the following flexible form: data storage and retrieval, preprocessing and standardization, dimensionality reduction (PCA), and clustering (K-means). Figure 2 shows an example of such a pipeline visualized using Orange, a Pythonic data mining

toolbox [\[30\]](#). The pipeline is flexible and capable of segmenting SKUs in the

unsupervised learning setting, which is crucial since the class values corresponding to an a priori partition of the SKUs are not known [31].

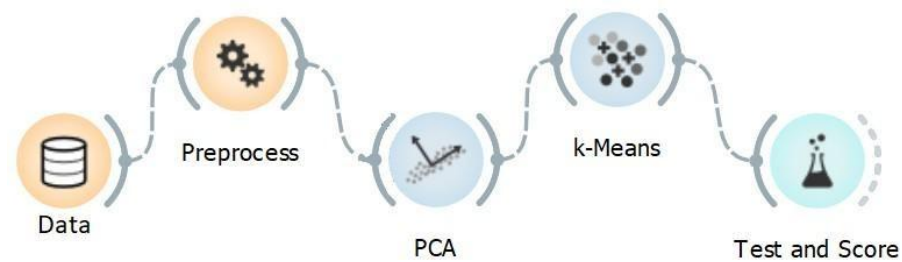


Figure 2. Typical data pipeline for SKU segmentation.

The popularity of K-means can be justified by the fact that more sophisticated clustering techniques such as mean shift [32] and density-based spatial clustering of applications with noise (DBSCAN) [33] do not allow for one to explicitly set the number of clusters (k in K-means). The finetuning of these algorithms is nontrivially performed by selecting “window size” for mean shift and “density reachability” for DBSCAN. Additionally, the algorithms do not perform well if SKU data contain convex clusters with significant differences in densities. The incorporation of PCA, on the other hand, allows for one to improve the clustering results for two main reasons. First, SKU features can share mutual information and correlate with each other. Second, it is natural for SKU data to contain some portion of noise due to the measurement or recording error. On that basis, PCA could be helpful as a tool in decorrelation and noise reduction [8].

Data Storage

The underlying data layout, including file format and schema, is an essential aspect in obtaining high computational performance and storage efficiency. The proposed framework for automated cluster analysis works on top of the Apache Parquet file format. Apache Parquet is an efficient, structured, compressed, binary file format that stores data in a column-oriented way instead of by row [34]. As a result, the values in each column are physically stored in contiguous memory locations, which provides the following crucial benefits:

- Data are stored by column instead of by row. Columnwise compression is spaceefficient because the specific compression techniques to a data type can be applied at the column level.
- Query processing is more efficient as the columns are stored together. This is because the query engine can fetch only specific column values, skipping the columns that are not needed for a particular query.
- Different encoding techniques can be applied to different columns, drastically reducing storage requirements without sacrificing query performance [35].

As a result, Parquet provides convenient and space-efficient data compression and encoding scheme with enhanced performance to operate with complex multidimensional and multitype data.

Feature Scaling

Feature scaling, also known as data normalization, is an essential step in data preprocessing. Machine-learning techniques such as K-means use distances between data points to determine their similarity that render them especially sensitive to feature scaling. The primary role of feature scaling in ML pipelines is to standardize the independent variables, so they are in a fixed range. If feature

scaling is not performed, K-means weigh variables with greater absolute values higher, severely undermining the clustering outcome [36].

The proposed framework for automated cluster analysis uses min–max scaling, the most straightforward method that scales features, so they are in the range of [0, 1]. The min–max scaling is performed according to the following formula:

$$x^0 = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x is an original value and x^0 is the normalized one.

Clustering

K-means is the most straightforward clustering method and, as the related work section revealed, is widely adopted for SKU segmentation purposes. K-means splits n observations (SKUs in the context of this study) into k clusters. Formally, given a set of SKUs $x = (x_1, x_2, \dots, x_n)$, where each SKU is a feature vector, K-means segments x into k sets $S = \{S_1, S_2, \dots, S_k\}$ minimizing the within-cluster sum of squares, also known as inertia $\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$.

The proposed framework for automated cluster analysis incorporates minibatch Kmeans, a variant of K-means clustering that operates with so-called minibatches to decrease computational time while still attempting to minimize inertia. Minibatches refer to the randomly and iteratively sampled subsets of the input data. This approach significantly reduces the computational budget required to converge to a nearly optimal solution [37]. Algorithm 1 demonstrates the key principle behind minibatch K-means. The algorithm iterates between two crucial steps. In the first step, *size* observations are sampled randomly to form a minibatch. Then, the minibatch is assigned to the nearest centroid that updates in the second step. In contrast to the standard K-means algorithm, the centroid updates are conducted per sample. These two steps are performed until convergence, or a predetermined stoppage criterion is met.

Algorithm 1 Minibatch K-means [37].

```

1: Given:  $k$ , minibatch size, iterations  $t$ , data set  $X$ 
2: Initialize each  $c \in C$  with an  $x$  picked randomly from  $X$ 
3:  $v \leftarrow 0$ 
4: for  $i = 1$  to  $t$  do
5:    $M \leftarrow$  size examples picked randomly from  $x$ 
6:   for  $x \in M$  do
7:      $d[x] \leftarrow f(C, x)$  . Store the center nearest to  $x$ 
8:   end for
9:   for  $x \in M$  do
10:     $x \leftarrow d[x]$  . Get cached center for this  $x$ 
11:     $v[c] \leftarrow v[c] + 1$  . Update per-center counts
12:     $\eta \leftarrow \frac{1}{v[c]}$  . Get per-center learning rate
13:     $c \leftarrow (1 - \eta)c + \eta x$  . Take gradient step

```

14: **end for** 15: **end for**

Dimensionality Reduction

Observations originated from natural phenomena or industrial processes frequently contain a large number of variables. A multitude of SKU features that can potentially impact operational and strategic decisions is a notable example. Therefore, the data analysis pipelines can benefit from dimensionality reduction in one form or another. In general, dimensionality reduction can be defined as the transformation of data into a low-dimensional representation, so that the obtained representation retains meaningful properties and patterns of the original data [38]. Principal component analysis (PCA) is the most widely adopted and theoretically grounded approach to dimensionality reduction. The proposed framework for automated cluster analysis incorporates PCA in order to decompose a multidimensional dataset of SKUs into a set of orthogonal components, such that the set of the obtained orthogonal components has a lower dimension, but is still capable of explaining a maximal amount of variance [39]. Computation for PCA starts with considering the dataset of n SKUs with m features. The SKU dataset can be represented as the following matrix:

$$X_n = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad (2)$$

where x_i is a i th row vector of length m . Thus, X_n is a $n \times m$ matrix whose columns represent features of a given SKU.

The sample means $\bar{x}_n$ and standard deviations σ_n can be defined with respect to the columns $x^{-n(j)} = \frac{\sum_{i=1}^n X_{n(ij)}}{n}$ and $\sigma_{n(j)} = \sqrt{(n-1)^{-1} \sum_{i=1}^n [X_{n(ij)} - \bar{x}_{n(j)}]^2}$. After that, the matrix can be standardized as follows:

$$Z_n = \frac{1}{\sigma_n} \begin{bmatrix} x_1 - \bar{x}_n \\ x - \bar{x}_n \\ \vdots \\ x_n - \bar{x}_n \end{bmatrix} \text{diag}(\sigma_n)^{-1}, \quad (3)$$

where $\text{diag}(\sigma_n)$ stands for a $m \times m$ matrix.

Covariance matrix Q_n can be defined as $Q_n = (n-1)^{-1} Z_n^T Z_n$. So, we can finally perform a standard diagonalization as follows:

$$Q_n = C_n^{-1} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} C_n, \quad (4)$$

where eigenvalues λ_i are arranged in descending order: $\lambda_i > \lambda_{i+1}$. Transformation matrix C_n is the orthogonal eigenvector matrix. The rows are the principal components of Q_n , and the value of λ_i represents the variance associated with the i th principal component [39].

Validation

Since the ground truth is not known, clustering results must be validated on the basis of an internal evaluation metric. Generally, inertia gives some notion of how internally coherent clusters are. However, the metric suffers from several disadvantages. For example, it assumes that clusters are convex and isotropic, which is not always the case with SKUs. Additionally, inertia is not a normalized metric, and Euclidean distances tend to inflate in high-dimensional spaces. Fortunately, the latest can be partially alleviated by PCA [40].

Unsupervised learning in general and clustering in particular approach the problem, which is biobjective by nature. On the one hand, data instances must be partitioned, such that points in the same cluster are similar to each other (this property is also known as cohesion). On the other hand, clusters themselves must be different from each other (a property known as separation). The silhouette coefficient is explicitly designed to evaluate the clustering results taking into account both cohesion and separation. The Silhouette coefficient is calculated for each sample and is composed of two components (a and b).

$$s = \frac{b - a}{\max(a, b)} \quad (5)$$

where a is the mean distance between a sample and all other points within the same cluster (measure of cohesion). b is the mean distance between a sample and all other points in the next nearest cluster (measure of separation) [41].

Parameter Optimization

Parameter optimization is a common problem in SKU segmentation, and a distinct issue from the process of dimensionality reduction and solving the clustering problem itself. Parameter optimization in the context of the unsupervised learning-based SKU segmentation is the process of determining the optimal number of clusters in a data set (k in K-means) and the optimal number of principal components (p in PCA) [42]. If appropriate values of k and p are not apparent from prior knowledge of the SKU properties, they must be defined somehow. That is where the elements of AutoML come into play. Grid search or a parameter sweep is the traditional way of performing parameter optimization [43]. The method performs an exhaustive search through a specified set of parameter space guided by the Silhouette score.

The proposed framework for automated cluster analysis is built on top of the pipeline that has two hyperparameters to be finetuned. Namely, the number of clusters $K = \{2, 3, 4, \dots\}$ and the number of principal components $P = \{1, 2, 3, \dots\}$. Two sets of hyperparameters K and P can be represented as a set of ordered pairs (k, p) , also known as Cartesian product $P \times K = \{(k, p) | k \in K \text{ and } p \in P\}$. Grid search then trains the pipeline with each pair (k, p) and outputs the settings that achieved the highest Silhouette score.

Grid search suffers from the curse of dimensionality, but since the parameter settings that it evaluates are independent of each other, the parallelization process is straightforward, allowing for one to leverage the specialized hardware and cloud solutions [44].

Implementation

The proposed framework for automated cluster analysis was implemented entirely in Python. The reading of the data from the Apache Parquet file format and all the manipulations with the data structures were performed using Pandas

[45], which was optimized for performance, with critical code paths written in such low-level programming languages as Cython and C [46].

The unsupervised learning components, including K-means and PCA, were implemented using scikit-learn, an open-source ML library for Python [47]. The scikit-learn implementation of K-means uses multiple processor cores benefiting from multiplatform shared-memory multiprocessing execution in Cython.

Additionally, data are processed in chunks of 256 samples in parallel, which results in a low memory footprint [48]. The scikitlearn implementation of PCA was based on [49]. The data visualization was performed using Seaborn, a Python data visualization library [50].

Results

The proposed framework was tested on the basis of a real-world dataset that contains SKU data of perishable grocery products at the pallet level. The dataset was provided by a third-party logistics provider from the Baltic region and contained data at the pallet level. The study was conducted in a reproducible manner. The data availability section at the end of this manuscript contains the link to the GitHub repository with the source code and original dataset.

Dataset Description

The dataset included 1218 rows that corresponded to the individual SKUs and 4 columns corresponding to SKU attributes or features. The features included unit price (EUR), number of units per pallet, gross weight of the loaded pallet (kg), and Pallet height (m).

Table 2 provides descriptive statistics, including measures of central tendency, dispersion, and quartiles. Figure 3 sheds light on the shape of the feature's distribution by presenting the box plots and kernel density estimation plots. The kernel density estimation plots are produced by applying kernel smoothing to estimate the probability density function of the empirical data. Such an illustration immediately provides critical insights into the data structure. Such features as “unit price”, “number of units”, and “gross weight” are heavy-tailed and multimodal (trimodal to be more specific).

It is also essential to check the presence of multicollinearity in the dataset. The Pearson correlation coefficient is the most straightforward approach to measuring the linear relationship between two features. The coefficient varies between -1 and $+1$, with correlations of -1 or $+1$ implying an exact linear relationship and 0 implying no correlation [51]. Table 3 contains Pearson's correlation coefficient and the corresponding p-values. The p-values indicate the probability of a pair of uncorrelated features having a correlation coefficient to be at least as extreme as the one calculated on the basis of data [51].

Table 2. Measures of central tendency, dispersion, and quartiles.

	Unit Price	Number of Units	Gross Weight	Height
mean	4.8	349.6	244.1	1.1
std	5.1	553.9	118.5	0.25
min	0.01	12.0	10.2	0.36
25%	1.1	72.0	165.8	0.9
50%	2.7	108.0	226.8	1.02
75%	5.6	384.0	293.8	1.05
max	19.8	2400.0	777.6	1.80

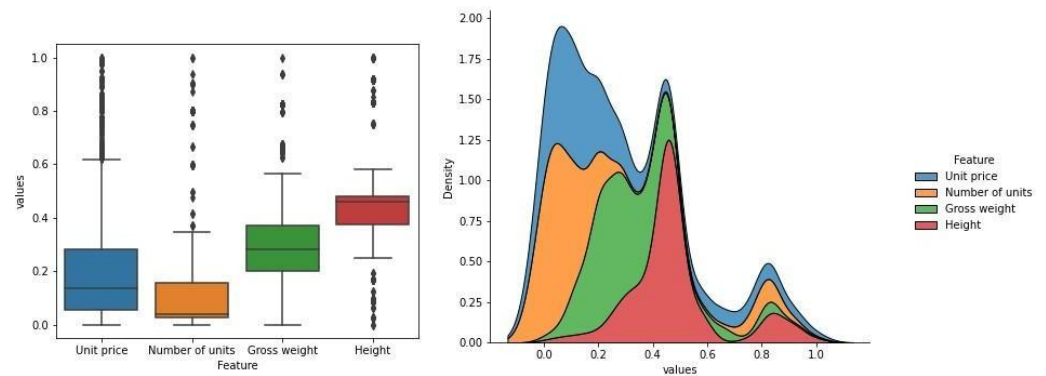


Figure 3. Box plots and kernel density estimation plots of the SKU features.

Even though K-means clustering is not severely affected by the presence of multicollinearity, correlation coefficients may provide a hint of whether PCA has the potential of improving clustering results. Besides that, the computation of correlation coefficients and plotting the features pairwise could reveal the data structure and provide valuable managerial insights. Figure A1 provides the bivariate view illustrating the relationship between two of the most correlated features, “gross weight” and “unit price”, and describing the strength of their relationship. Figure A2 illustrates the pairwise relationships in a dataset. The diagonal plots correspond to a univariate distribution plot demonstrating the marginal distribution of each feature.

Table 3. Pearson correlation coefficient with corresponding p-values.

	Unit Price	Number of Units	Gross Weight	Height
Unit price	1.0	−0.44 (0.0)	0.61 (0.0)	0.02 (0.45)
Number of units	−0.44 (0.0)	1.0	−0.31 (0.0)	0.30 (0.0)
Gross weight	0.61 (0.0)	−0.31 (0.0)	1.0	0.23 (0.0)

Numerical Experiment

In order to test the proposed framework, the grid search was performed across the search space produced by the Cartesian product of $K = \{2, 3, 4, \dots, 9\}$ and $P = \{0, 2, 3\}$, where $p = 0$ corresponded to the case where PCA was not applied. Figure 4 shows

Silhouette scores of the pipeline with each pair of parameters.

The pipeline with three principal components and five clusters demonstrated the highest performance (Silhouette score of 0.699). Additionally, five clusters appeared to be optimal even when PCA had not been applied. However, such a pipeline performed notably worse (Silhouette score of 0.662). The pipeline with two principal components and three clusters could be considered as an alternative with the second best performance (Silhouette score of 0.690). Table A1 contains the Silhouette score values for every pair of pipeline parameters obtained through the grid-search procedure.

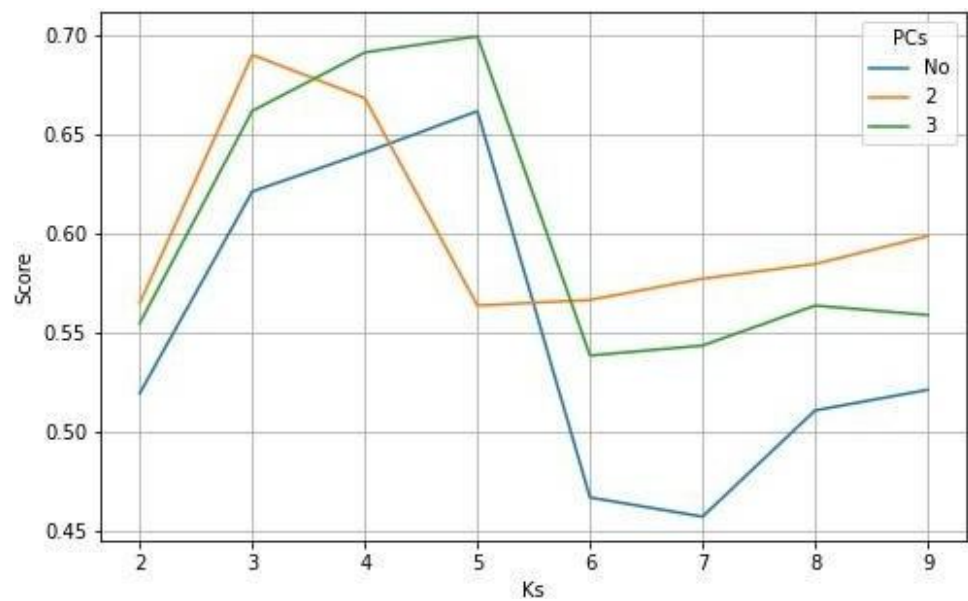


Figure 4. Silhouette scores calculated across the Cartesian product.

In a real-world setting, decision making could be driven not only by the performance metric, but also by a plethora of other factors and considerations. For example, even though the SKU segmentation with three principal components and five clusters yielded the highest Silhouette score, the priority may be given to the second best solution (two principal components and three clusters) because fewer clusters are easier to manage. This relatively simple case reveals an additional advantage of the proposed framework. Namely, it provides and ranks multiple solutions, but it is up to a manager to make a final decision considering other factors (potentially non-numerical) initially not taken into account by the algorithm. Figure A3 shows the pairwise view of the clustering results. The diagonal plots demonstrate the marginal distribution of each feature within each cluster. On the basis of visualizations, the potential end user could conclude that the produced SKU clusters are homogeneous to be treated under common inventory management or material handling policy.

Discussion

This chapter discusses our research contributions and the technological advantages of the proposed implementation. It also discusses several use cases and provides managerial implications.

Research Contributions

The idea of prioritizing items for management attention dates back to the 1950s [5], and the first attempt to cluster SKUs dates back to the 1990s [7]. As time progresses, new clustering techniques are applied to this longstanding problem. Nowadays, the most common techniques include hierarchical clustering, K-means, fuzzy C-means, and K-means combined with PCA.

Even though hierarchical clustering is simple to implement and interpret at the managerial level, it does not scale well because of the computational complexity. The algorithm is also biased towards large clusters and tends to break them even if it leads to suboptimal segmentation. The method is also sensitive to noise and outliers [19].

K-means is appealing in many aspects, including computational and memory efficiency, and interpretability in low-dimensional cases [20]. However, the

method requires the number of clusters to be specified in advance. In business applications, the a priori specification of the number of clusters is a severe drawback because it requires an additional effort to parametrize and finetune the model. Additionally, K-means performs poorly on datasets that contain clusters with unequal sizes or densities, and is also sensitive to both noise and outliers [20].

Fuzzy C-means is an alternative to K-means that treats a cluster as a fuzzy set, allowing for SKUs to belong to more than one cluster. This approach may have theoretical importance. However, the fact that each SKU can belong to more than one cluster is a clear disadvantage in the business context. Since supply chains operate under pressure from both supply and demand, businesses have to make critical decisions quickly and confidently. Therefore, the fuzziness of SKU segments does not provide any helpful flexibility for decision making at scale, and introduces unnecessary layers of complexity to decision making [27].

In order to overcome the mentioned shortcomings, preprocessing and dimensionality reduction techniques such as PCA are applied to SKU data prior to clustering. The incorporation of PCA allows for one to improve the clustering results through decorrelation and noise reduction [38]. However, even though the combination of K-means and PCA helps to overcome some critical problems, it further worsens the parametrization and finetuning problem because the potential user must choose the number of principal components and the number of clusters. Our work closes this research gap by proposing to automate parametrization and finetuning. We propose an AutoML-based framework for SKU segmentation that leverages grid search along with the computationally efficient and parallelism-friendly implementations of clustering and dimensionality reduction algorithms. The following subsections shed light on the technological advantages of the proposed solution and managerial implications.

Technological Contributions

The proposed framework for automated cluster analysis can work with SKU features beyond that utilized by a classical ABC approach. The framework operates in an unsupervised learning setting, which is crucial since class values corresponding to an a priori partition of the SKUs are not known. The optimal number of clusters in a dataset and the optimal number of principal components are determined through the grid search in a fully automated fashion using the Silhouette score as a performance estimation metric.

In a real-world setting, decision making could be less straightforward and incorporate unquantifiable metrics. In this regard, the critical advantage of the framework is that the grid search examines every possible solution within the search space and ranks them in accordance with the Silhouette score. Therefore, the end user can select a solution that is aligned with some potentially unquantifiable managerial insights, but is not necessarily best according to the performance estimation criteria. Additionally, the framework is equipped with data-visualization capabilities, which give end users the ability to diagnose the segmentation validity, choose among alternative solutions, and effectively communicate results to the critical stakeholders.

The proposed framework is strongly associated with the so-called “no free lunch” theorem. According to the theorem, there cannot exist a superior algorithm to any competitive one across the domain of all possible problems [52]. In this regard, the question of whether the proposed framework is inferior or superior to the potential alternative is meaningless, since its performance would depend on the data generation mechanism and distribution of feature values. Nevertheless, the

following properties should be highlighted as the indisputable practical advantages of the proposed approach:

- **Automatism.** Such labor-intensive and time-consuming steps as parameter optimization and finetuning are fully automated and do not require the participation of data scientists or other human experts.
- **Parallelism.** The proposed framework for automated cluster analysis incorporates minibatch K-means, a computationally cheap variant of K-means clustering. The implementation of minibatch K-means uses multiple processor cores benefiting from multiplatform shared-memory multiprocessing execution in Cython. Additionally, data are processed in chunks of 256 samples in parallel. The parallelization process for the grid search is straightforward as well, which allows for one to leverage the specialized hardware and cloud solutions.
- **Big-data-friendliness.** The proposed framework operates on top of the Apache Parquet file format, an efficient, structured, compressed, partition-friendly, and column-oriented data format. The reading of the data from the Apache Parquet file format and all the manipulations with the data structures were performed using Pandas, which was optimized for performance.
- **Interpretability.** Data-visualization capabilities allow for a potential user to diagnose the segmentation validity, spot outliers, gain critical insights regarding the data structure, and effectively communicate results to the critical stakeholders.
- **Universality.** The optimal number of clusters in a dataset and the optimal number of principal components are usually not apparent from prior knowledge of the SKU properties. Since the proposed framework performs an exhaustive search through a specified set of parameters, it may be applied to any SKU dataset with numeric features or even extended to similar domains, for example, customer or supplier segmentation.

The latter could be postulated as a promising direction for future research.

Managerial Implications

Treating thousands of units individually is overwhelming for businesses of all sizes, and beyond anyone's managerial effort and attention. Additionally, it may be simply computationally unfeasible to invoke a sophisticated forecasting technique or inventory policy at the granular SKU level. Segmentation implies focus, and allows for one to define a manageable number of SKU groups to facilitate strategy, and enhance various operational planning and material handling activities.

A classical example involves segmenting SKUs by margin and volume, which allows for a manager to focus on high-volume and high-margin products [6]. So, a very-high-cost, high-margin product line with volatile demand would benefit from a flexible and responsive supply chain. For this line, a company can afford to spend more on transportation to meet demand and keep the high fill rate. On the other hand, low-cost everyday items with stable demand can take advantage of a slow and economical supply chain to minimize costs and increase margins. Additionally, segmenting raw materials may help in applying a better sourcing strategy, and segmentation based on demand patterns and predictability may help in estimating the supply-chain resilience and manage risks accordingly. The numerical experiment demonstrated in our manuscript is an example of SKU segmentation based on the physical property of goods at the pallet level. Such

segmentation is aimed at deriving SKU clusters that can be treated under common material handling policy.

Other common use cases include a selection of an appropriate inventory control policy [7], categorizing SKUs by the demand patterns [9], optimal allocation under the limited storage capacity [10], selecting an appropriate supply-chain strategy [9], and evaluating the environmental impacts of inventory throughout its life cycle [11].

Real-world supply chains and operations are very dynamic and subject to change. The company's product portfolio may change, and it may not know what the margins and demand patterns will be, or some SKUs are more challenging to handle due to their physical properties and require longer lead times. In these cases, cluster analysis must be conducted once again with the updated data that potentially contains new features. That is where the true power of our solution comes into play. Since the optimal number of clusters in a dataset and the optimal number of principal components are usually not apparent from the SKU properties, they must be finetuned. However, parametrization and finetuning are labor-intensive and time-consuming steps that involve the participation of highly qualified experts like data scientists or machine-learning engineers. The proposed framework for automated cluster analysis fully automates finetuning. Therefore, the precious time of the qualified labor force can be reallocated to other corporate tasks. Additionally, the computational cheapness and parallelism-friendliness of the proposed framework entail scalability, which is crucial as a business grows. Besides that, our solution is supplemented with data-visualization capabilities that allow for a potential user to diagnose the segmentation validity, spot outliers, gain critical insights regarding the data structure, and effectively communicate results to critical stakeholders.

Conclusions

A retailer may carry thousands or even millions of SKUs. Besides the volume, SKU data can be high-dimensional. Given the data volumes and the multitude of potentially important dimensions to consider, it could be computationally challenging to manage each SKU individually. In order to address this problem, a framework for automated cluster analysis was proposed. The framework incorporates minibatch K-means clustering, principal component analysis, and grid search for parameter tuning. It operates on top of the Apache Parquet file format, an efficient, structured, compressed, column-oriented data format; therefore, it is expected to be efficient for real-world SKU segmentation problems of high dimensionality.

The proposed framework was tested on the basis of a real-world dataset that contained SKU data at the pallet level. Since the framework is equipped with data-visualization tools, the potential user can diagnose the segmentation validity. For example, using the pairwise view of the features after clustering, the user can conclude that the produced SKU clusters are homogeneous enough to be treated under common inventory management or material handling policy. The data visualization tools allow for one to spot outliers, gain critical insights regarding the data structure, and effectively communicate results to critical stakeholders. It is also essential to highlight a few limitations of our solution. First, K-means clustering, which constitutes a core of the proposed framework, segments data exclusively on the basis of Euclidean distance and is thereby not applicable to categorical data. Therefore, the incorporation of variations in K-means capable of working with non-numerical data can be postulated as a useful extension of our solution and a promising direction for future research. K-modes [53] and K-

prototypes [54] are preliminary candidates for this extension. Second, PCA is restricted to only a linear map, which may lead to a substantial loss of variance (and information) when the dimensionality is reduced drastically. Autoassociative neural networks, also known as autoencoders, have the potential to overcome this issue through their ability to stack numerous nonlinear transformations to reduce input into a low-dimensional latent space [55]. Therefore, exploring autoencoders may also be considered the second promising direction for future research. Since the proposed framework performs an exhaustive search through a specified set of parameters, it may be applied to similar domains, for example, customer or supplier segmentation, which is the third promising extension and direction for future research.

Funding: this research received no external funding.

Data Availability Statement: the complete implementation of the framework, including the automated parameter optimization and data, is available in the GitHub repository <https://github.com/Jackil1993/autologistics>, accessed on 1 November 2022.

Conflicts of Interest: the author declares no conflict of interest.

Appendix A

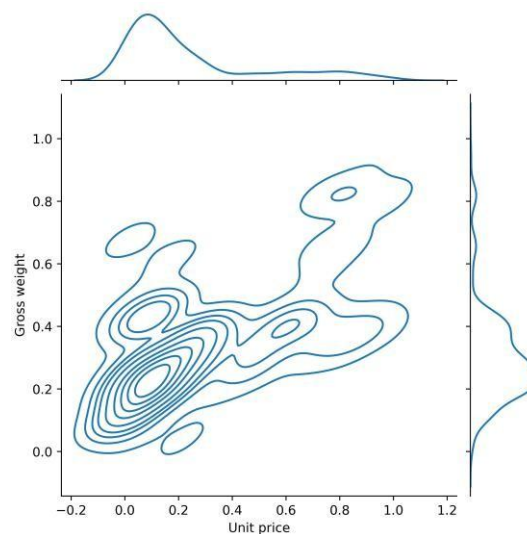


Figure A1. Two of the most correlated features, “gross weight” and “unit price”, with bivariate and univariate graphs.

Appendix B

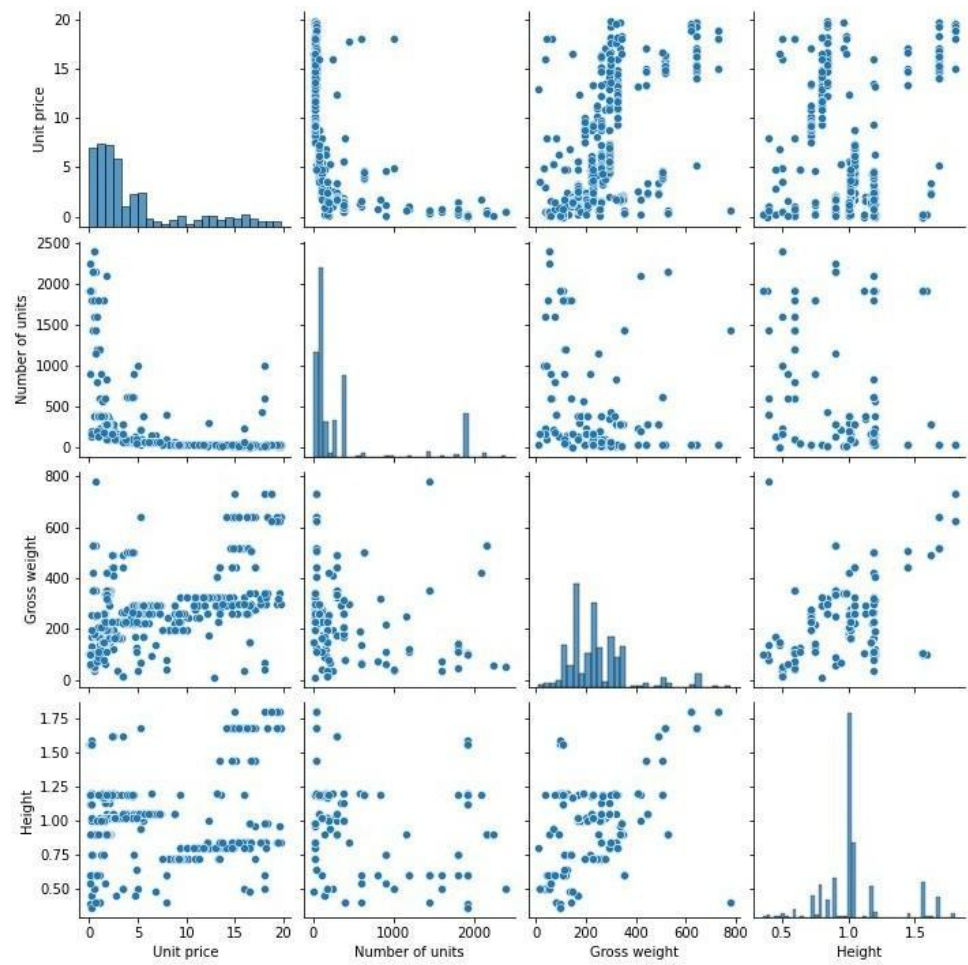


Figure A2. Pairwise relationships in a dataset. Diagonal plots correspond to a univariate distribution plot demonstrating the marginal distribution of each feature.

Appendix C

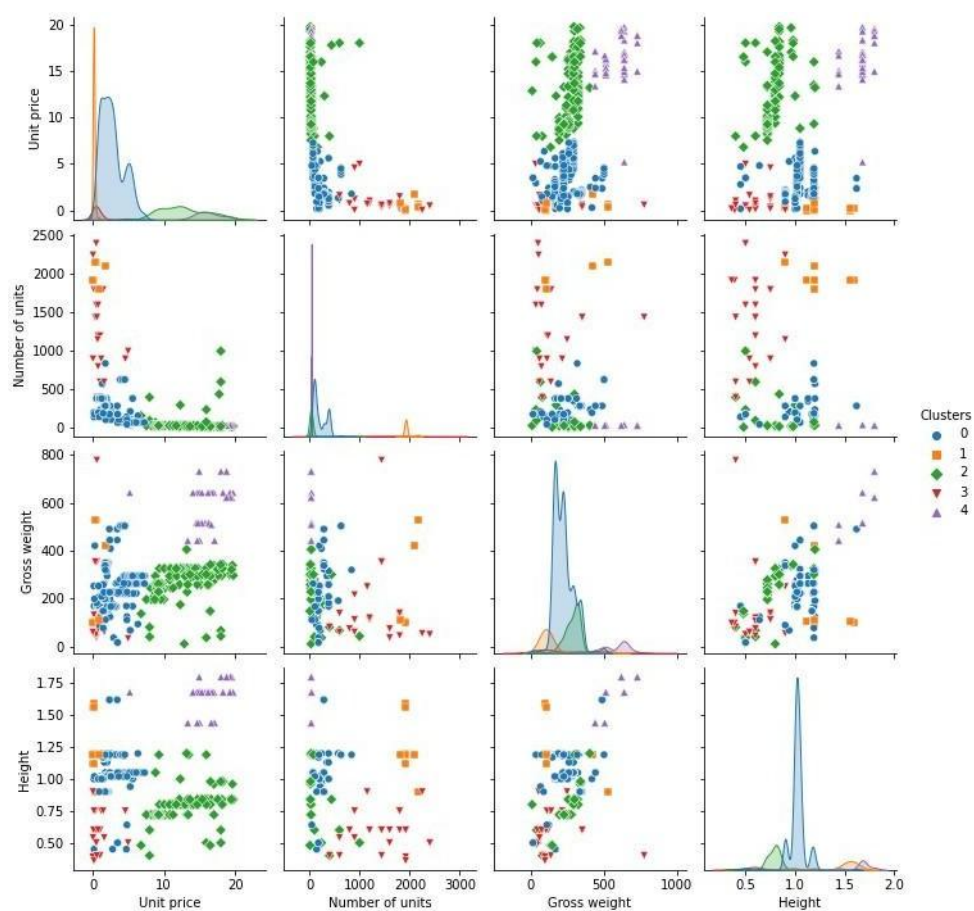


Figure A3. Pairwise view of features after clustering. Diagonal plots demonstrate the marginal distribution of each feature within each cluster (numbered from 0 to 4).

Appendix D

Table A1. Silhouette scores values for every pair of pipeline parameters obtained through the grid-search procedure.

PCs	Ks	Score
No	2	0.5245
	3	0.6196
	4	0.6408
	5	0.6616
	6	0.4670
	7	0.4573
	8	0.5109
	9	0.5212
2	2	0.5650
	3	0.6900
	4	0.6683
	5	0.5637

	6	0.5664
	7	0.5772
	8	0.5846
	9	0.5987
3	2	0.5547
	3	0.6618
	4	0.6913
	5	0.6994
	6	0.5385
	7	0.5435
	8	0.5637
	9	0.5589

References

1. Phadnis, S.S.; Sheffi, Y.; Caplice, C. Scenario Creation in Supply Chain Contexts. In *Strategic Planning for Dynamic Supply Chains*; Springer International Publishing: Cham, Switzerland, 2022; pp. 85–120. [\[CrossRef\]](#)
2. Jackson, I. Deep Reinforcement Learning for Supply Chain Synchronization. In Proceedings of the Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2022. [\[CrossRef\]](#)
3. US Department of Commerce. Manufacturing and Trade Inventories and Sales, Main Page, US Census Bureau. 2022. Available online: <https://www.census.gov/mtis/index.html> (accessed on 15 October 2022).
4. Summerlin, R.; Powell, W. Effect of Interactivity Level and Price on Online Purchase Intention. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 652–668. [\[CrossRef\]](#)
5. Juran, J. Universals in Management Planning and Control. *Manag. Rev.* **1954**, *43*, 748.
6. Fisher, M.; Raman, A. *The New Science of Retailing*; Harvard Business Review Press: Boston, MA, USA, 2010.
7. Ernst, R.; Cohen, M.A. Operations related groups (ORGs): A clustering procedure for production/inventory systems. *J. Oper. Manag.* **1990**, *9*, 574–598. [\[CrossRef\]](#)
8. Jackson, I.; Avdeikins, A.; Tolujevs, J. Unsupervised Learning-Based Stock Keeping Units Segmentation. In *Reliability and Statistics in Transportation and Communication*; Kabashkin, I., Yatskiv (Jackiva), I., Prentkovskis, O., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 603–612.
9. Fisher, M.; Hammond, J.; Obermeyer, W.; Raman, A. Configuring a Supply Chain to Reduce the Cost of Demand Uncertainty. *Prod. Oper. Manag.* **1997**, *6*, 211–225. [\[CrossRef\]](#)
10. Das, L.; Carrel, A.L.; Caplice, C. Pack size effects on retail store inventory and storage space needs. *INFOR Inf. Syst. Oper. Res.* **2021**, *59*, 465–494. [\[CrossRef\]](#)
11. Guinée, J.; Heijungs, R. Introduction to Life Cycle Assessment. In *Sustainable Supply Chains*; Springer International Publishing: Cham, Switzerland, 2016; pp. 15–41. [\[CrossRef\]](#)
12. Thomson Reuters Streetevents. WMT—Q4 2018 Wal Mart Stores Inc Earnings Call. 2018. Available online: https://corporate.walmart.com/media-library/document/q4fy18-earnings-webcast-transcript/_proxyDocument?id=00000161-d2c0-dfc5-a76b-f3f01e430000 (accessed on 15 October 2022).
13. Big Commerce. Amazon Statistics You Should Know: Opportunities to Make the Most of America’s Top Online Marketplace. 2018. Available online: <https://www.bigcommerce.com/blog/amazon-statistics/#a-shopping-experience-beyond-compare> (accessed on 15 October 2022).
14. Flores, B.E.; Whybark, D.C. Multiple Criteria ABC Analysis. *Int. J. Oper. Prod. Manag.* **1986**, *6*, 38–46. doi: 10.1108/eb054765. [\[CrossRef\]](#)
15. Jackson, I. Neuroevolutionary Approach to Metamodeling of Production-Inventory Systems with Lost-Sales and Markovian Demand. In *Lecture Notes in Networks and Systems*; Springer International Publishing: Cham, Switzerland, 2020; pp. 90–99. [\[CrossRef\]](#)
16. Cohen, M.C.; Gras, P.E.; Pentecoste, A.; Zhang, R. Clustering Techniques. In *Demand Prediction in Retail*; Springer International Publishing: Cham, Switzerland, 2022; pp. 93–114. [\[CrossRef\]](#)
17. Srinivasan, M.; Moon, Y.B. A comprehensive clustering algorithm for strategic analysis of supply chain networks. *Comput. Ind. Eng.* **1999**, *36*, 615–633. [\[CrossRef\]](#)

18. Bacchetti, A.; Plebani, F.; Saccani, N.; Syntetos, A. Empirically-driven hierarchical classification of stock keeping units. *Int. J. Prod. Econ.* **2013**, *143*, 263–274. [\[CrossRef\]](#)
19. Hennig, C.; Meila, M.; Murtagh, F.; Rocci, R. (Eds.) *Handbook of Cluster Analysis*; Chapman and Hall/CRC: London, UK, 2015. [\[CrossRef\]](#)
20. Wierzbach, S.; Kłopotek, M. *Modern Algorithms of Cluster Analysis*; Studies in Big Data; Springer International Publishing: Cham, Switzerland, 2019.
21. Canetta, L.; Cheikhrouhou, N.; Glardon, R. Applying two-stage SOM-based clustering approaches to industrial data analysis. *Prod. Plan. Control.* **2005**, *16*, 774–784. [\[CrossRef\]](#)
22. Wu, S.D.; Aytac, B.; Berger, R.T.; Armbruster, C.A. Managing Short Life-Cycle Technology Products for Agere Systems. *Interfaces* **2006**, *36*, 234–247. [\[CrossRef\]](#)
23. Egas, C.; Masel, D.T. Determining Warehouse Storage Location Assignments Using Clustering Analysis. In Proceedings of the 11th IMHRC Conference, Milwaukee, WI, USA, 21–24 June 2010.
24. Ozturk, Z.K.; Cetin, Y.; Isik, Y.; Cicek, Z.E. Demand Forecasting with Clustering and Artificial Neural Networks Methods: An Application for Stock Keeping Units. In *Springer Proceedings in Mathematics and Statistics*; Springer International Publishing: Cham, Switzerland, 2021; pp. 355–368. [\[CrossRef\]](#)
25. Keskin, G.A.; Ozkan, C. Multiple Criteria ABC Analysis with FCM Clustering. *J. Ind. Eng.* **2013**, *2013*, 1–7. [\[CrossRef\]](#)
26. Kucukdeniz, T.; Erkal, S. Integrated Warehouse Layout Planning with Fuzzy C-Means Clustering. In Proceedings of the International Conference on Intelligent and Fuzzy Systems—INFUS 2022, Izmir, Turkey, 19–21 July 2022. [\[CrossRef\]](#)
27. Memon, K.H.; Lee, D.H. Generalised fuzzy c-means clustering algorithm with local information. *IET Image Process.* **2017**, *11*, 1–12. [\[CrossRef\]](#)
28. Yang, C.L.; Quyen, N.T.K. Constrained clustering method for class-based storage location assignment in warehouse. *Ind. Manag. Data Syst.* **2016**, *116*, 667–689. [\[CrossRef\]](#)
29. Bandyopadhyay, S.; Thakur, S.S.; Mandal, J.K. Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: Benefit for the society. *Innov. Syst. Softw. Eng.* **2020**, *17*, 45–52. [\[CrossRef\]](#)
30. Crt Gorup; Demšar, J.; Curk, T.; Erjavec, A.; Hočvar, T.; Milutinović, M.; Možina, M.; Polajnar, M.; Toplak, M.; Starič, A.; et al. Orange: Data Mining Toolbox in Python. *J. Mach. Learn. Res.* **2013**, *14*, 2349–2353.
31. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2nd ed.; Data-Centric Systems and Applications; Springer: Heidelberg, Germany; New York, NY, USA, 2011.
32. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [\[CrossRef\]](#)
33. Tran, T.N.; Drab, K.; Daszykowski, M. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemom. Intell. Lab. Syst.* **2013**, *120*, 92–96. [\[CrossRef\]](#)
34. Vohra, D. Apache Parquet. In *Practical Hadoop Ecosystem*; Apress: New York, NY, USA, 2016; pp. 325–335. [\[CrossRef\]](#)
35. Floratou, A. Columnar Storage Formats. In *Encyclopedia of Big Data Technologies*; Springer International Publishing: Cham, Switzerland, 2019; pp. 464–469. [\[CrossRef\]](#)
36. Han, J.; Pei, J.; Kamber, M.; Safari, A.O.M.C. *Data Mining: Concepts and Techniques*, 3rd ed.; Elsevier: Amsterdam, The Netherlands, 2011; OCLC: 1112917381.
37. Sculley, D. Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web—WWW'10, Raleigh, NC, USA, 26–30 April 2010; ACM Press: New York, NY, USA, 2010. [\[CrossRef\]](#)
38. van der Maaten, L.; Postma, E.O.; van den Herik, J. *Dimensionality Reduction: A Comparative Review*; Tilburg University: Tilburg, The Netherlands, 2009.
39. Lippi, V.; Ceccarelli, G. Incremental Principal Component Analysis: Exact Implementation and Continuity Corrections. In Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, Prague, Czech Republic, 29–31 July 2019; SCITEPRESS—Science and Technology Publications: Setúbal, Portugal, 2019. [\[CrossRef\]](#)
40. de Amorim, R.C.; Hennig, C. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Inf. Sci.* **2015**, *324*, 126–145. doi: 10.1016/j.ins.2015.06.039. [\[CrossRef\]](#)
41. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [\[CrossRef\]](#)
42. Pelleg, D.; Moore, A.W. X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00), Stanford, CA, USA, 29 June–2 July 2000; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000; pp. 727–734.
43. Chicco, D. Ten quick tips for machine learning in computational biology. *BioData Min.* **2017**, *10*, 35. [\[CrossRef\]](#)
44. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
45. McKinney, W. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; pp. 56–61. [\[CrossRef\]](#)
46. Pandas Development Team. pandas-dev/pandas: Pandas. 2020. Available online: <https://zenodo.org/record/7223478#.Y3HSduRBxPY> (accessed on 1 November 2022). [\[CrossRef\]](#)
47. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

48. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. In Proceedings of the ECML PKDD Workshop: Languages for Data Mining and Machine Learning, Prague, Czech Republic, 23 September 2013; pp. 108–122.
49. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2007**, *77*, 125–141. [[CrossRef](#)]
50. Waskom, M.L. seaborn: Statistical data visualization. *J. Open Source Softw.* **2021**, *6*, 3021. [[CrossRef](#)]
51. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [[CrossRef](#)]
52. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
53. Chaturvedi, A.; Green, P.E.; Carroll, J.D. K-modes Clustering. *J. Classif.* **2001**, *18*, 35–55. [[CrossRef](#)]
54. Ji, J.; Bai, T.; Zhou, C.; Ma, C.; Wang, Z. An improved k-prototypes clustering algorithm for mixed numeric and categorical data. *Neurocomputing* **2013**, *120*, 590–596. [[CrossRef](#)]
55. Guo, X.; Liu, X.; Zhu, E.; Yin, J. Deep Clustering with Convolutional Autoencoders. In *Neural Information Processing*; Springer International Publishing: Cham, Switzerland, 2017; pp. 373–382. [[CrossRef](#)]

Demand-Based Product Classification Using K-Means with Intermittency Metrics

Ariska Nur Anggraini^{1,*}, Amali¹, Muhammad Syaibani Anwar¹

¹ Universitas Pelita Bangsa, Indonesia

* Corresponding author: Ariska Nur Anggraini, Universitas Pelita Bangsa, Indonesia
✉ ariskanur112@gmail.com

Copyright: © 2026 by the authors

Received: 2 April 2026 | Revised: 12 April 2026 | Accepted: 25 April 2026 | Published: 29 April 2026

Abstract

Inventory management at multi-SKU distribution companies becomes complex when most products have unstable and intermittent demand patterns. At PT JJA, procurement is still reactive without the use of historical patterns, while the previous approach generally relied on aggregate indicators such as average sales so that it has not been able to comprehensively capture temporal dynamics. This study aims to group products based on temporal demand patterns using K-Means Clustering in 11,988 transactions for the 2020–2025 period which are processed into 261 products through monthly aggregation, with features of average sales, coefficient of variation (CV), zero_month_ratio, Average Demand Interval (ADI), and trends. The results showed four optimal clusters ($k = 4$) with a Silhouette Score of 0.62 and an unbalanced distribution, where one cluster dominated 240 products. The values of zero_month_ratio (>0.80), ADI up to >12 months, and CV up to >3.5 show intermittent demand patterns and long-tail structures. The study confirms that the integration of temporal features (ADI, zero_month_ratio, CV, and trend) transforms the representation of demand from static aggregates to dynamic structures, while linking segmentation results with more adaptive procurement strategies to reduce the risk of overstock and understock.

Keywords: k-means clustering; product segmentation; procurement of goods

To cite this article: Anggraini, A. N., Amali, A., & Anwar, M. S. (2026). Demand-Based Product Classification Using K-Means with Intermittency Metrics. *Edumatic: Jurnal Pendidikan Informatika*, 10(1), 280–289. <https://doi.org/10.29408/edumatic.v10i1.34435>

INTRODUCTION

Inventory management on a multi-SKU system faces complexity when the majority of products exhibit intermittent and unstable demand patterns. This condition often leads to improper procurement decisions, such as overstock on products that are rarely sold and stock shortages on the products needed. These differences in characteristics make a uniform management approach less than optimal in practice (Wisner et al., 2023). As well as having an impact on excess stock that binds working capital and running out of stock which reduces customer satisfaction (Heizer et al., 2023). Therefore, an approach that is able to group products based on demand characteristics is needed that is more representative. However, there is no consensus in the literature on the most appropriate indicators to represent simultaneous intermittent demand patterns on multi-SKU systems.

Product demand on a multi-SKU system is generally not constant, but rather shows temporal patterns such as fluctuations, trends, and no-sales periods. This condition reflects intermittent demand that is difficult to capture using conventional aggregate-based approaches.



Although methods such as K-Means are computationally effective, the use of aggregate features such as sales averages has the potential to eliminate the time dimension, making interpretations of product characteristics biased (Kolassa et al., 2023; Sanguri et al., 2024). This condition often leads to improper procurement decisions, such as overstock on products that are rarely sold and stock vacancies on the products needed. This problem is also reinforced by the finding that traditional forecasting methods that ignore temporal dynamics tend to result in inaccurate predictions and suboptimal inventory decisions (Wang & Ning, 2026; Wang & Minner, 2024).

PT JJA still implements a reactive procurement pattern (make to order) without the support of systematic demand prediction, even though historical sales data stores important information related to demand patterns. To overcome this, this study uses an unsupervised learning-based data mining approach with the K-Means Clustering algorithm to group products based on the similarity of demand characteristics. This method was chosen because it is efficient for large-scale data, although it has limitations such as sensitivity to centroid initialization and cluster count determination (Pramudya et al., 2026), and applied to support decision-making based on unstable demand patterns (Ezugwu et al., 2023). However, the use of K-Means in the previous literature is still generally limited to aggregate data-based grouping or static features, so it has not fully explored the temporal character of requests in the context of multi-SKUs.

While previous study has emphasized the use of machine learning and clustering techniques in demand analysis and supply chain management, existing approaches still tend to be partial in accommodating temporal dynamics comprehensively. A study by Giannopoulos et al. (2023) It is still at the level of general model synthesis without the specification of operational temporal features, so methodologically it does not provide guidance for the representation of time-based data and has implications for segmentation that is less sensitive to temporal variations. On the other hand Tarragó et al. (2023), use temporal data in a homogeneous context, which limits its methodological scope and has implications for low generalizations to heterogeneous multi-SKU systems. Further, Keskin and Taşkın (2024) Develop an artificial intelligence-based inventory classification at a conceptual level without formulating time variability as a key feature, so that the resulting model tends to be static in response to demand fluctuations. Meanwhile, Vlachos and Reddy (2025) Focuses on an overview of machine learning architecture and trends, which methodologically creates a gap with operational implementation and has implications for the difficulty of direct application in decision-making. In addition, Figuera et al. (2024) emphasizing the validation of clustering statistics without linking them to the context of business decisions, so that the evaluation carried out does not guarantee the managerial relevance of cluster results.

Drawing on previous studies, several scientific and practical limitations remain insufficiently addressed in an integrated manner. From a scientific perspective, most research continues to rely on aggregate indicators such as average sales and transaction frequency, without simultaneously incorporating intermittent metrics like Average Demand Interval (ADI) and Coefficient of Variation (CV). Moreover, classification outcomes are generally confined to analytical purposes and are rarely linked to inventory management strategies. From a practical standpoint, PT JJA still applies a make-to-order procurement approach without leveraging historical data as a basis for product segmentation, resulting in demand patterns not being fully utilized to support more precise and data-driven decision-making.

This study uses an unsupervised learning-based data mining approach with the K-Means Clustering algorithm to group products based on similar demand patterns. This approach integrates temporal indicators such as average sales, coefficient of variation (CV), number of periods without sales, Average Demand Interval (ADI), and demand trends, so that it is able to represent demand dynamics more comprehensively. This study is important because it connects

clustering techniques in data mining with inventory management based on demand characteristics, which so far generally still uses simple aggregate indicators and has not been able to capture simultaneous intermittent and dynamic demand patterns.

The simultaneous use of time-based demand indicators, average sales, fluctuation rates, sales consistency, number of months without sales, and demand trends, as well as the integration of grouping results into procurement recommendations differentiates this approach from conventional segmentation. This method goes beyond aggregate indicators by including temporal dimensions as an explicit representation of demand patterns that were not previously combined in a single grouping process, resulting in a more representative product classification to support decision-making on a multi-SKU system.

This study aims to apply the K-Means Clustering algorithm in product segmentation based on temporal demand patterns on PT JJA's sales data, as well as identify cluster characteristics and formulate procurement strategies based on segmentation results. The contribution of this study is to provide a data mining approach that can be used as part of a decision support system to support more targeted inventory management.

METHOD

This study uses an exploratory unsupervised learning design with a data mining approach using the K-Means algorithm for product segmentation based on temporal demand patterns at PT JJA. K-Means was chosen for efficiency on large data, although it has limitations in handling outliers and determining the optimal number of clusters early in analysis. Methods such as DBSCAN, hierarchical clustering, and GMM were excluded due to uneven density data, high computational complexity, and normal distribution assumptions.

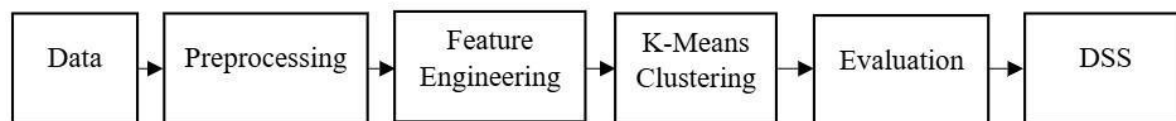


Figure 3. Research methodology pipeline

Figure 3 presents the systematic research flow from pre-processing, feature engineering, grouping of K-Means, cluster evaluation, to the utilization of results in procurement DSS. The object of the study is PT JJA with 11,988 transactions for the 2020–2025 period, including the COVID-19 period. The data shows the distribution of long tail, a small percentage of the product is sold very often, while the majority is rarely sold (see Table 1). This imbalance is maintained as a natural character of a multi-SKU system. Data was collected through observation, interviews, and documentation, and then cleaned, standardized, and normalized using Z-score due to K-Means' sensitivity to the Euclidean scale. Outliers are maintained as a representation of intermittent surge in demand. There are no lost values and months without transactions are recorded as zero. The data is then converted into a monthly sales matrix per product to support the analysis of temporal demand patterns in inventory management.

Table 1. Monthly sales data per product

Product Code	Jan-2020	Feb-2020	Mar-2020	...	Oct-2025	Nov-2025	Dec 2025
AA-1510	0	0	1	...	0	0	0
AA-1520	0	0	0	...	0	0	0
AA-1530	0	0	0	...	0	0	0
AA-1540	0	0	0	...	0	0	0
AA-1550	5	0	0	...	0	0	0

The derived attributes obtained from processing the data in Table 1 into statistical measures such as average demand, coefficient of variation (CV), number of active months, proportion of zero-sales months, Average Demand Interval (ADI), and trend are presented in Table 2, capturing the underlying demand characteristics and serving as key inputs for the clustering analysis.

Table 2. Product attribute formation results

Product Code	Total Qty	Mean Monthly	CV	Active Months	Zero Month Ratio	ADI	Trend
AA-1510	7	0.097	3.49	6	0.92	12	0.0004
AA-1520	7	0.097	3.04	7	0.90	10.29	0.0021
AA-1530	9	0.125	2.96	8	0.89	9.00	0.0007
...	...	...	...	...	...	...	...

Evaluation of the number of clusters using the Elbow and Silhouette Score method. Elbow assesses the SSE's decline against k-variation to find a slowdown point, while Silhouette measures the quality of cluster separation in the range of -1 to 1. The determination of k is optimal based on the combination of both. Data processing is done in Python (Google Colab) with Excel support. K-Means runs on k2=10, k-means++ initialization, max_iter 300, and random_state. The result is in the form of product grouping based on temporal demand patterns as the basis for procurement strategy recommendations.

RESULTS AND DISCUSSION

Results

The dataset consists of 11,988 transactions that have been processed into 261 products with a representation of monthly sales over 72 periods. Most products have a low sales frequency, while a small percentage of products have a higher sales frequency. Feature engineering produces zero_month_ratio values with an average of more than 0.80 on most products. The ADI value is in the range of about 3 to more than 12 months. The value of the CV is in the range of ± 0.8 to more than 3.5.

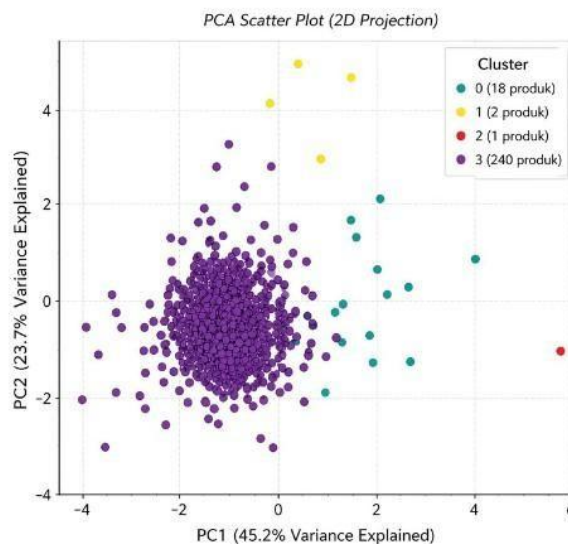


Figure 1. Cluster separation visualization (pca projection)

Determination of the number of clusters resulted in $k = 4$ as the configuration used. The Sum of Squared Error (SSE) value decreased to $k = 4$ and slowed down thereafter. The highest

Silhouette Score value was obtained at $k = 4$ which was about 0.62, compared to $k = 3$ (~ 0.54) and $k = 5$ (~ 0.58). The cluster distribution consisted of 240 products in Cluster 3, 18 products in Cluster 0, 2 products in Cluster 1, and 1 product in Cluster 2.

Figure 1 contains a visualization of cluster separation using two-dimensional PCA projection. The projection results in a clear cluster separation between the dominant Cluster 3 and the cluster with a small number of members. The points in Clusters 0, 1, and 2 are in a more scattered position and are far from the center of Cluster 3 in the projection plane of the two main components. The difference in position between clusters represents the difference in demand characteristics in each cluster during the data observation period.

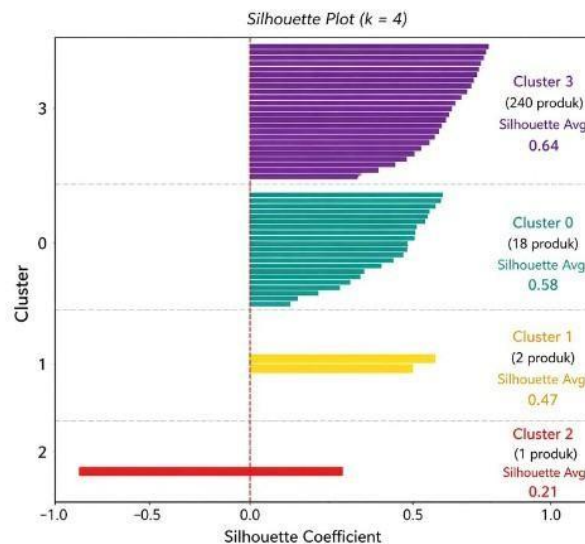


Figure 2. Silhouette plot

Figure 2 contains a silhouette plot showing most of the products in Cluster 3 with positive and relatively high silhouette values, reflecting the good grouping consistency of most of the cluster members during the segmentation process. Meanwhile, the products in the small cluster still have a positive silhouette value, so even though the number of members is small, the separation formed remains structurally valid, consistent, and clearly separate based on the proximity of the demand feature characteristics between the products.

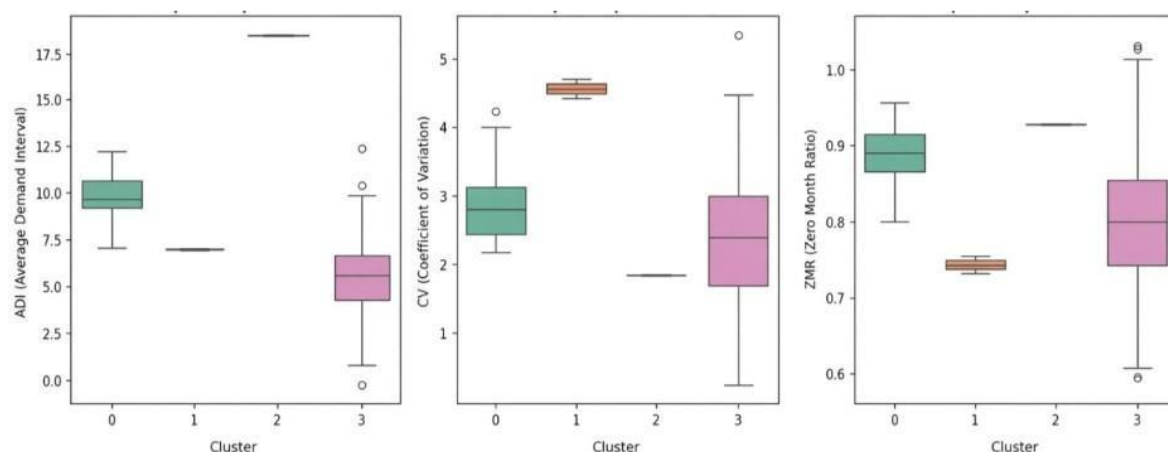


Figure 3. Distribution of adi, cv, and zmr values in each cluster

Figure 3 contains the distribution of ADI, CV, and ZMR values in each K-Means result cluster. There are differences in characteristics between clusters. Cluster 2 has the highest ADI and ZMR values, with very rare demand patterns, long intervals between transactions, and many months without sales (very rare demand). Cluster 1 has the highest CV value, with extreme and unstable demand fluctuations. Meanwhile, Cluster 3 with the most product members has a distribution of ADI, CV, and ZMR at the intermediate level, which represents the dominant moderate intermittent demand pattern in the multi-SKU system. This visualization confirms that each cluster has a structurally different demand pattern and cannot be managed with a uniform procurement approach.

Cluster 3 consists of 240 products. Clusters 0, 1, and 2 consist of 18, 2, and 1 products, respectively. Products in cluster 3 have ADI and CV values in the medium to high range. Meanwhile, products in clusters with smaller members have higher ADI values or larger CV values ($CV > 3$), with sparse, fluctuating, unpredictable, and unstable demand patterns during the observation period. These findings confirm that the demand structure in a multi-SKU system is not homogeneous, but rather dominated by products with intermediate patterns as well as a small number of products with extreme behavior, which cannot be managed with a uniform procurement approach. Therefore, adaptive and segmentation-based procurement strategies are needed to accommodate differences in demand characteristics between products more effectively and efficiently.

The distribution of the number of cluster members is patterned by a long-tail distribution, most of the products are concentrated in a single cluster (Cluster 3). A small number of products are in clusters with extreme characteristics. This condition is commonly found in multi-SKU inventory systems. The centroids-based nature of K-Means leads to the tendency to form one large cluster that represents a general pattern. Products with extreme behavior separate into small clusters. This is not a weakness, but rather the ability of K-Means to isolate demand patterns that deviate from the majority.

Table 3. Distribution of the number of products in each cluster

Cluster	Count
0	18
1	2
2	1
3	240

Table 3 presents the distribution of the number of products in each cluster. Cluster 3 consists of 240 products, while Cluster 0, Cluster 1, and Cluster 2 consist of 18, 2, and 1 products, respectively, with the predominance of products with the same characteristics and the presence of a small number of products with very different demand patterns, in the range of frequency, variation, interval, and consistency of sales between time periods.

The stability test was carried out by rerunning the K-Means algorithm using several different random seed values. The results obtained were consistent in the grouping of products in the same cluster, especially in the dominant Cluster 3. All products in Cluster 3 remained in the same cluster each time the iteration process was rerun. The products in Clusters 0, 1, and 2 also did not move between clusters even though the initial centroid used was different. Based on these findings, the cluster structure formed was proven to be stable, robust, and not dependent on initial initialization. This strengthens the validity of the segmentation results that have been obtained previously.

Table 4 contains the differences in demand pattern characteristics between K-Means product clusters at PT JJA based on the average ADI, CV, ZMR, and trends. Clusters with high ADI and ZMR represent products with long demand intervals and many months without

transactions, reflecting very rare demand. Clusters with the highest CV reflect extreme and unpredictable demand fluctuations. Meanwhile, clusters with ADI, CV, and ZMR values at the medium level but with the most product members describe the dominant moderate intermittent demand pattern in multi-SKU systems. The difference in trend values between clusters confirms that there is a variation in demand trends over time, so the results of this segmentation are relevant as the basis for determining procurement strategies and inventory control.

Table 4. Average value of adi, cv, zmr, and trend per k-means result cluster at pt jja

cluster	Avg ADI	Avg CV	Avg ZMR	Avg trend
0	9.87	3.12	0.91	0.004
1	6.21	4.85	0.73	-0.012
2	18.44	2.02	0.96	0.001

Clusters with a small number of members still have a good silhouette value, which reflects a consistent separation between groups. This condition indicates that the cluster structure has a good level of interpretability, where each cluster is formed based on clear differences in demand characteristics, both in terms of frequency, variation, and transaction patterns between observation periods.

Table 5. Examples of product clustering results

Product Code	Item Name	Cluster	Cluster Type	Procurement Strategy
AA-1510	HONEYWELL EDA 52	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1520	Water Wind Glue Shot Caulking Gun	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1530	CBA-R07-S07PAR Cable RS232	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1540	DS3678-SR Rugged Scanner Kit	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1550	310P1.25 NAC Impact Sockets	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1560	Makita GA 4030	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1570	Tube Package APAR Tonata ABC 1 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1580	Tube Package APAR Tonata HFC 6 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring
AA-1590	Tube Package APAR Tonata HFC 1 Kg	3	Moderate/ Special	Selective Procurement/ Monitoring

Table 5 presents examples of products grouped by cluster. This table contains information about product codes, item names, cluster labels, cluster types, and procurement strategies. The products shown in table 5 are from Cluster 3. The final result of this study is the collection of product grouping data that can be used as a basis for evaluating demand patterns and formulating inventory procurement strategies in the company, as well as providing an analytical basis to support more accurate, additive, and demand-based decision-making for each product in the context of complex multi-SKU inventory management.

Discussion

This study shows that the formation of highly unbalanced clusters, with the dominance of products in one large cluster and a small number of products in extreme clusters, has important theoretical significance. This structure has to do with the natural character of multi-SKU systems, where most products are on a medium demand pattern that appears relatively rarely but not completely sporadic, while few products are actually highly volatile or highly stable. This kind of distribution is in line with the character of long-tail demand distribution in the product portfolio, so grouping is naturally concentrated in one large cluster with several extreme clusters (Bi et al., 2023). This condition explains the limitations of the uniform procurement approach that is classically assumed in the demand homogeneity-based inventory model. The dominance of Cluster 3 that houses most of the products is not solely a numerical result, but reflects that the majority of items in the distribution system have a moderate intermittent demand character. These products do not have a high frequency of demand, but they are also not completely random, so conceptually form large groups with similar temporal characteristics. This dominance shows that in the classification of demand, the intermediate pattern is the main representation of the portfolio, while the extreme pattern represents only a small percentage of the product.

The high ADI value in some clusters has the conceptual implication that the distance between requests is a more decisive dimension than the volume of demand itself. The frequency of demand emergence becomes a more relevant indicator than the aggregate sales volume, as products with small volumes but long intervals have different behaviors than short-interval products, a concept that is reflected in the ADI-CV approach to the classification of demand patterns and changes in demand behavior in SKUs that have a Long demand interval (Neu et al., 2024). This expands the classic perspective of inventory management which has so far emphasized the magnitude of demand rather than the rhythm of its emergence.

K-Means builds clusters by defining centroids as the center of data mass. This algorithm minimizes intra-cluster spacing through iterative optimization, resulting in mutually exclusive partitions (Kaur et al., 2024). Farahnakian et al. (2023) asserts that the main objective function of K-Means is to minimize within-cluster variance so that separation between clusters occurs naturally through such optimization. In contrast to the commonly used transaction frequency-based grouping or static aggregate value approach, this study expands the representation of data by integrating temporal features such as zero_month_ratio, ADI, coefficient of variation, and demand trends. This integration transforms the data structure from a mere transaction intensity to a dynamic representation of demand behavior, so that the grouping reflects not only the size of demand, but also the patterns of regularity and fluctuations in demand across each product (Mudgal, 2026). Thus, the use of this temporal feature provides a richer perspective on data mining-based product segmentation, particularly in the context of multi-SKU data with non-uniform demand patterns.

Previous study as shown that the use of machine learning and clustering techniques in demand analysis and supply chain management has evolved, but it still has limitations in comprehensively capturing temporal dynamics. In line with Giannopoulos et al. (2023), placing intermittent requests at the level of general model synthesis without specification of operational temporal features, so that methodologically it does not provide guidance for time-based data representation and has implications for segmentation that is less sensitive to temporal variations. Meanwhile, Tarragó et al. (2023) adopt temporal data in homogeneous contexts without addressing the implications of those context's limitations, so that the generalization of the model to heterogeneous multi-SKU systems becomes weak. In contrast to Keskin and Taşkın (2024) propose an artificial intelligence-based inventory classification at a conceptual level without formulating time variability as a key feature, which has implications for classification models that tend to be static to real demand fluctuations. Complete Vlachos and

Reddy (2025), focuses on the integration of analytical models and machine learning architectures without explaining the methodological implications of the gap with operational practices, resulting in limited direct application in decision-making. This study is also in line with the results of the study Figuera et al. (2024) emphasizing statistical validation of clustering without identifying methodological weaknesses of the approach in the context of business decisions, so the managerial relevance of cluster results is not guaranteed.

In contrast to static aggregate-based approaches such as total or average sales, this study uses temporal features (zero_month_ratio, ADI, and CV) so that demand is understood as a dynamic process with a structure of regularity, irregularity, and emptiness, in line with the feature-based forecasting literature for intermittent demand (Li et al., 2023). This shift has the theoretical implication that product classification is not enough based on demand volume alone, but it is necessary to consider temporal dimensions to more accurately identify intermittent patterns, especially in probabilistic forecasting that is directly related to inventory decisions (Park & Rogetzer, 2022). Thus, this study expands the classification of classical inventory from an aggregate approach to segmentation based on temporal dynamics.

This study contributes in three principal dimensions. First, it develops a temporal feature-based segmentation framework for multi-SKU systems by integrating zero_month_ratio, ADI, coefficient of variation, and trend to capture intermittent demand dynamics. Second, it reconceptualizes K-Means from a purely numerical clustering method into an analytical instrument for revealing latent demand structures shaped by temporal variability and uncertainty. Third, it links segmentation outcomes to operational decision-making, enabling differentiated inventory strategies such as safety stock determination and order frequency optimization. Collectively, this approach advances a shift from aggregate-based classification toward temporally grounded segmentation, with novelty residing in the transformation of demand representation into dynamic, behavior-oriented structures.

CONCLUSION

This study shows that K-Means Clustering in PT JJA's sales data produces four demand pattern clusters with uneven distribution, where Cluster 3 dominates as many as 240 products, reflecting the long-tail characteristics of the multi-SKU system. These findings confirm that most products fall into the category of low or unstable demand. The main contribution of this study is the use of temporal indicators such as variation coefficient, month ratio without sales, ADI, and trends that enrich segmentation compared to the sales volume-based approach, thus providing a more informative representation of demand patterns for procurement decision-making. Practically, these results support the design of a more adaptive procurement strategy based on the characteristics of each cluster. However, this study is still limited to internal historical data without considering external variables such as macroeconomic conditions. Further study can develop this approach with time series-based models such as LSTM or ensemble learning to improve the accuracy of demand analysis.

REFERENCES

- Bi, S., He, L., & Teo, C. P. (2023). Taming the Long Tail: The Gambler's Fallacy in Intermittent Demand Management. *Manufacturing & Service Operations Management*, 25(5), 1692–1710. <https://doi.org/10.1287/msom.2023.1217>
- Ezugwu, A. E., Ikotun, A. M., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences Elsevier*, 622, 178–210. <https://doi.org/10.1016/j.ins.2022.11.139>
- Farahnakian, F., Nicolas, F., Farahnakian, F., Nevalainen, P., Sheikh, J., Heikkonen, J., & Raduly-Baka, C. (2023). A comprehensive study of clustering-based techniques for

- detecting abnormal vessel behavior. *Remote Sensing*, 15(6), 1477. <https://doi.org/10.3390/rs15061477>
- Figuera, P., Cuzzocrea, A., & García Bringas, P. (2024). Clustering Validation Inference. *Mathematics*, 12(15), 2349. <https://doi.org/10.3390/math12152349>
- Giannopoulos, P. G., Dasaklis, T. K., Tsantilis, I., & Patsakis, C. (2025). Machine learning algorithms in intermittent demand forecasting: a review. *International Journal of Production Research*, 1-43. <https://doi.org/10.1080/00207543.2025.2578701>
- Heizer, J., Render, B., & Munson, C. (2023). *Operations Management: Sustainability and Supply Chain Management, Global Edition* (14th ed.). Pearson Higher.
- Kaur, A., Kumar, Y., & Sidhu, J. (2024). Exploring meta-heuristics for partitional clustering: methods, metrics, datasets, and challenges. *Artificial Intelligence Review*, 57(10), 287. <https://doi.org/10.1007/s10462-024-10920-1>
- Keskin, S., & Taşkın, A. (2024). Inventory classification with artificial intelligence: Conceptual framework. *Journal of Decision Analytics and Intelligent Computing*, 4(1), 136-142. <https://doi.org/10.31181/jdaic10029092024k>
- Kolassa, S., Rostami-Tabar, B., & Siemsen, E. (2023). *Demand Forecasting for Executives and Professionals* (1st ed.). Chapman and Hall. <https://doi.org/10.1201/9781003399599>
- Li, L., Kang, Y., Petropoulos, F., & Li, F. (2023). Feature-based intermittent demand forecast combinations: accuracy and inventory implications. *International Journal of Production Research*, 61(22), 7557–7572. <https://doi.org/10.1080/00207543.2022.2153941>
- Mudgal, P. (2026). Clustering of temporal and visual data: Recent advancements. *Data*, 11, 7. <https://doi.org/10.3390/data11010007>
- Neu, Z., Hicks, B., & Gopsill, J. (2024). Operating minimally intelligent agent-based manufacturing systems across the Average demand Interval–coefficient of variation (ADI-CV) demand state space. *Production & Manufacturing Research*, 12(1), 2323479. <https://doi.org/10.1080/21693277.2024.2323479>
- Park, D., & Rogetzer, P. (2022). Timing intermittent demand with time-varying order-up-to levels. *European Journal of Operational Research*, 303(3), 1126–1136. <https://doi.org/10.1016/j.ejor.2022.03.019>
- Pramudya, R. I., Kurniawan, T. A., Candra, M. H., Onn, C. W., & Dissanayake, K. K. (2026). Advancing unsupervised clustering: A systematic review of hybrid K-means and metaheuristic optimization algorithms in data mining. *Computer Science Review*, 61, 100972. <https://doi.org/10.1016/j.cosrev.2026.100972>
- Sanguri, K., Patra, S., Nikolopoulos, K., & Punia, S. (2024). Intermittent demand, inventory obsolescence, and temporal aggregation forecasts. *International Journal of Production Research*, 62(5). <https://doi.org/10.1080/00207543.2023.2199435>
- Tarragó, R., Bottoms, L., & Iglesias, X. (2023). Temporal demands of elite fencing. *Plos one*, 18(6), e0285033. <https://doi.org/10.1371/journal.pone.0285033>
- Vlachos, I., & Reddy, P. G. (2025). Machine learning in supply chain management: systematic literature review and future research agenda. *International Journal of Production Research*, 63(16), 5987-6016. <https://doi.org/10.1080/00207543.2025.2466062>
- Wang, C., & Ning, G. (2026). Multi-criteria inventory classification considering demand stability. *Scientific Reports*, 30(16). <https://doi.org/10.1038/s41598-026-42590-0>
- Wang, Y., & Minner, S. (2025). Data-driven multi-location inventory placement in digital commerce. *Computers & Industrial Engineering*, 200, 110842. <https://doi.org/10.1016/j.cie.2024.110842>
- Wisner, J. D., Tan, K.-C., & Leong, G. K. (2023). *Principles of Supply Chain Management: A Balanced Approach* (6th ed.). Cengage.



DECLARACIÓN Y AUTORIZACIÓN

Yo, FrANCESCA Valentina Castro Perrone con C.C: # 0932159288 autor/a del trabajo de titulación: **Optimización del stock de los productos de Toalco mediante un algoritmo K-means** previo a la obtención del título de **Licenciada en 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. _____

Nombre: Francesca Valentina Castro Perrone

C.C: 0932159288



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Jostyn Josue Parraga Rosales** con C.C: # 0957800824 autor/a del trabajo de titulación: **Optimización del stock de los productos de Toalco mediante un algoritmo K-means** previo a la obtención del título de **Licenciado en 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. _____

Nombre: **Jostyn Josue Parraga Rosales**

C.C: 0957800824

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TEMA Y SUBTEMA:	Optimización del stock de los productos de Toalco mediante un algoritmo K-means		
AUTOR(ES)	Franccesca Valentina Castro Perrone Jostyn Josue Parraga Rosales		
REVISOR(ES)/TUTOR(ES)	Carrera Buri, Félix Miguel		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Economía y Empresa		
CARRERA:	Negocios Internacionales		
TÍTULO OBTENIDO:	Licenciada/o en Negocios Internacionales		
FECHA DE PUBLICACIÓN:	22 de agosto de 2026	No. DE PÁGINAS:	158
ÁREAS TEMÁTICAS:	Inteligencia Artificial, Machine Learning, Optimización del Stock		
PALABRAS CLAVES/ KEYWORDS:	Inventario, Segmentación de productos, Análisis de Componentes Principales, K-Means, Aprendizaje no supervisado, Optimización de stock, Gestión de inventarios.		
RESUMEN/ABSTRACT: La gestión correcta de inventarios es parte fundamental en la competitividad de las empresas en la actualidad, donde la innovación y la automatización de procesos mediante el análisis de datos se ha convertido en una herramienta efectiva para mejorar la clasificación de los productos, identificando las variables que influyen directamente en el comportamiento de este. Pese a ello, el entendimiento y la utilización de conceptos de aprendizaje automático, como lo son el Análisis de Componentes Principales y el algoritmo K-Means para la segmentación de productos, siguen siendo un desafío. La clasificación de los productos y la identificación de las variables que afectan su rotación dentro del inventario son de principal importancia, ya que gracias a esto las empresas pueden adoptar estrategias adecuadas de aprovisionamiento y reducir los riesgos de sobrestock y desabastecimiento. La presente investigación se enfoca en el desarrollo de una metodología basada en el Análisis de Componentes Principales y el algoritmo K-Means para la segmentación de productos de inventario, en la cual se espera lograr una agrupación representativa de los productos según su comportamiento de demanda, costo y rotación. La aplicación del Análisis de Componentes Principales y del algoritmo K-Means para esta segmentación se justifica debido a que la gestión de inventarios constituye uno de los procesos operativos más determinantes en la rentabilidad y la eficiencia de las empresas, al representar una parte significativa de su capital de trabajo.			
ADJUNTO PDF:	☒ SI	☐ NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593996941753 +593 93 908 2057	E-mail: francesca.castro@cu.ucsg.edu.ec jostyn.parraga@cu.ucsg.edu.ec	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Nombre: Freire Quintero César Enrique Teléfono: +593 99 009 0702 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):			