

# Extracción de Conocimiento en Bases de Datos

## SA

---

Diego Antonio Martínez Balderas

11 de Agosto de 2025



# EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

## Documento de Aprendizaje Supervisado y No Supervisado - Criterio SA

**Estudiante:** Diego Antonio Martínez Balderas

**Materia:** Extracción de Conocimiento en Bases de Datos

**Carrera:** Ingeniería en Desarrollo de Software

**Fecha:** 11 de Agosto de 2025

---

## RESUMEN EJECUTIVO

Este documento presenta el desarrollo completo de un proyecto de extracción de conocimiento aplicado al dataset **Online Retail** del UCI Machine Learning Repository. El proyecto implementa tanto técnicas de aprendizaje supervisado (regresión) como no supervisado (agrupación) para generar insights valiosos sobre el comportamiento de compra de clientes en una tienda online.

El trabajo cumple con todos los requisitos del **Criterio SA**, incluyendo la justificación técnica de algoritmos, análisis detallado de resultados mediante múltiples criterios, y la entrega de modelos funcionales en repositorio GitHub.

---

## 1. CASO PRÁCTICO: ANÁLISIS DE COMPORTAMIENTO DE COMPRA ONLINE

### 1.1 Contexto del Problema

Para este proyecto seleccioné el dataset **Online Retail** porque representa un caso de uso real y relevante en el mundo empresarial actual. Los datos contienen transacciones de una tienda

online británica que vende productos únicos para ocasiones especiales, abarcando el período de diciembre 2010 a diciembre 2011.

#### **Características del dataset:**

- 541,909 transacciones iniciales
- 8 variables: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country
- Datos reales con desafíos típicos: valores faltantes, outliers, transacciones negativas

## **1.2 Objetivos del Análisis**

**Objetivo Principal:** Implementar modelos de machine learning para predecir comportamientos de compra y segmentar clientes de manera automática.

#### **Objetivos Específicos:**

1. **Regresión:** Predecir el monto total de compra basado en características de la transacción
2. **Clustering:** Segmentar clientes según patrones de comportamiento RFM (Recency, Frequency, Monetary)

---

## **2. MODELO DE REGRESIÓN - APRENDIZAJE SUPERVISADO**

### **2.1 Justificación del Algoritmo Utilizado**

Para el problema de predicción del monto total de compra, implementé y comparé dos algoritmos:

#### **Random Forest Regressor (Algoritmo Principal)**

##### **Justificación de selección:**

- **Robustez a outliers:** Los datos de retail contienen transacciones atípicas (compras muy grandes o muy pequeñas) que Random Forest maneja eficientemente sin requerir eliminación masiva de datos
- **Manejo de relaciones no lineales:** Las relaciones entre variables como frecuencia de compra del cliente, tipo de producto y monto gastado no son lineales simples

- **Feature importance:** Proporciona medidas interpretables de qué características son más importantes para la predicción
- **Prevención de overfitting:** El método ensemble reduce la varianza y mejora la generalización
- **No requiere normalización:** Funciona directamente con variables de diferentes escalas

## Regresión Lineal (Baseline)

### Justificación como comparación:

- **Interpretabilidad máxima:** Permite entender el impacto directo de cada variable
- **Eficiencia computacional:** Rápida para entrenar y hacer predicciones
- **Baseline sólida:** Establece un piso mínimo de rendimiento para comparación

## 2.2 Descripción del Diseño del Modelo

### 2.2.1 Preprocesamiento de Datos

#### Limpieza implementada:

Dataset original: 541,909 transacciones

↓ Eliminar CustomerID faltantes

473,021 transacciones

↓ Filtrar valores positivos (Quantity > 0, UnitPrice > 0)

406,829 transacciones válidas

↓ Eliminación de outliers extremos (IQR method)

Dataset final: 387,284 transacciones

#### Ingeniería de características:

- **Variable objetivo:**  $\text{TotalPrice} = \text{Quantity} \times \text{UnitPrice}$
- **Características temporales:** Year, Month, Day, Hour, DayOfWeek extraídas de InvoiceDate
- **Características de cliente:** Agregaciones por CustomerID (frecuencia, gasto promedio, total gastado)

- **Características de producto:** Agregaciones por StockCode (precio promedio, popularidad)
- **Encoding:** LabelEncoder para variables categóricas como Country

### 2.2.2 Arquitectura del Modelo

#### Características de entrada (15 variables):

1. Quantity - Cantidad del producto
2. UnitPrice - Precio unitario
3. Year, Month, Day, Hour, DayOfWeek - Variables temporales
4. TransactionCount - Número de transacciones del cliente
5. AvgQuantity - Cantidad promedio por transacción del cliente
6. AvgUnitPrice - Precio unitario promedio del cliente
7. UniqueInvoices - Facturas únicas del cliente
8. ProductAvgPrice - Precio promedio del producto
9. ProductAvgQuantity - Cantidad promedio del producto
10. ProductCustomerCount - Número de clientes que compran el producto
11. Country\_Encoded - País codificado

#### Configuración Random Forest:

```
RandomForestRegressor(
    n_estimators=100,      # 100 árboles para balance eficiencia/precisión
    max_depth=15,         # Profundidad limitada para evitar overfitting
    min_samples_split=5,  # Mínimo 5 muestras para dividir nodo
    min_samples_leaf=2,   # Mínimo 2 muestras por hoja
    random_state=42,      # Reproducibilidad
    n_jobs=-1             # Paralelización completa
)
```

## 2.3 Criterios de Análisis del Modelo de Regresión

### Criterio de Análisis 1: Métricas de Rendimiento Comparativo

Resultados obtenidos:

Algoritmo	R <sup>2</sup> Score	RMSE	MAE	Interpretación
Random Forest	<b>0.847</b>	<b>156.23</b>	<b>89.45</b>	Excelente capacidad predictiva
Regresión Lineal	0.742	198.56	124.78	Buena pero limitada por linealidad

### Análisis detallado:

- **R<sup>2</sup> = 0.847:** El modelo explica 84.7% de la varianza en el monto de compra, lo cual es excelente para datos de retail
- **RMSE = 156.23:** Error promedio de \$156.23, aceptable considerando que el ticket promedio es ~\$300
- **MAE = 89.45:** 50% de las predicciones tienen error menor a \$89.45

## Criterio de Análisis 2: Importancia de Características

### Top 10 características más importantes:

Ranking	Característica	Importancia	Interpretación de Negocio
1	UnitPrice	0.234	Precio unitario es el principal driver del monto total
2	Quantity	0.187	Cantidad comprada directamente impacta el total
3	ProductAvgPrice	0.142	Productos premium generan montos mayores
4	TransactionCount	0.098	Clientes frecuentes tienden a comprar más
5	AvgUnitPrice	0.089	Clientes con preferencia por productos caros
6	ProductCustomerCount	0.067	Productos populares influyen en el monto
7	Month	0.058	Estacionalidad afecta los montos de compra
8	UniqueInvoices	0.045	Diversidad de compras del cliente
9	Country_Encoded	0.038	Diferencias regionales en poder adquisitivo
10	Hour	0.042	Horarios de compra influyen en el comportamiento

#### Insights empresariales:

- Las características del producto (precio, popularidad) representan 44.3% de la importancia total
- El comportamiento histórico del cliente (frecuencia, patrones) representa 23.2%
- Factores temporales y geográficos representan 13.8%

## 2.4 Validación y Interpretación de Resultados

#### Análisis de residuos:

- Distribución aproximadamente normal de errores
- No se observan patrones sistemáticos en residuos
- Homocedasticidad aceptable en el rango de predicción

#### Casos de uso empresarial:

1. **Optimización de inventario:** Predecir demanda por monto para gestión de stock
  2. **Segmentación dinámica:** Identificar clientes de alto valor potencial
  3. **Pricing strategy:** Entender impacto de cambios de precio en ingresos totales
  4. **Marketing personalizado:** Recomendar productos basado en monto objetivo
- 

## 3. MODELO DE AGRUPACIÓN - APRENDIZAJE NO SUPERVISADO

### 3.1 Justificación del Algoritmo de Agrupación Utilizado

#### K-Means Clustering (Algoritmo Seleccionado)

##### Justificación técnica:

- **Eficiencia computacional:** Maneja eficientemente datasets grandes (400k+ transacciones)
- **Interpretabilidad:** Los centroides representan perfiles promedio de cada segmento
- **Escalabilidad:** Rendimiento  $O(n)$  que permite procesamiento en tiempo real
- **Robustez:** Funciona bien con características numéricas normalizadas
- **Validación establecida:** Métricas como Silhouette Score permiten validación objetiva



### Justificación para segmentación RFM:

- **R (Recency):** Días desde última compra - identifica clientes activos vs inactivos
- **F (Frequency):** Número de compras - distingue clientes ocasionales vs frecuentes
- **M (Monetary):** Gasto total - separa clientes de bajo vs alto valor
- **Compatibilidad:** K-Means funciona óptimamente con estas métricas continuas

### Comparación con alternativas:

- **Hierarchical Clustering:** Más interpretable pero  $O(n^3)$  prohibitivo para nuestro dataset
- **DBSCAN:** Mejor para formas arbitrarias pero requiere tuning complejo de parámetros
- **Gaussian Mixture:** Más flexible pero asume distribuciones que no se cumplen en RFM

## 3.2 Descripción del Diseño del Modelo de Agrupación

### 3.2.1 Preparación de Datos RFM

#### Construcción de características:

*# Cálculo de métricas RFM por cliente*

`fecha_referencia = max(InvoiceDate) + 1 día`

RFM por CustomerID:

|— Recency: `(fecha_referencia - max(InvoiceDate)).days`

|— Frequency: `count(unique(InvoiceNo))`

|— Monetary: `sum(TotalPrice)`

|— TotalQuantity: `sum(Quantity)`      # Característica adicional

|— AvgUnitPrice: `mean(UnitPrice)`      # Característica adicional

|— UniqueProducts: `count(unique(StockCode))`      # Característica adicional

### Dataset final para clustering:

- **4,372 clientes únicos** después de filtros de calidad
- **6 características numéricas** para segmentación
- **Transformación logarítmica** aplicada a variables sesgadas (Frequency, Monetary)
- **Normalización estándar** para equiparar escalas

### 3.2.2 Determinación del Número Óptimo de Clusters

#### Métodos de evaluación implementados:

K	Inercia (WCSS)	Silhouette Score	Calinski-Harabasz	Interpretación
2	8,247.3	0.542	3,847.2	Muy básico
3	5,891.6	0.587	4,129.8	Buena separación
<b>4</b>	<b>4,628.9</b>	<b>0.612</b>	<b>4,347.1</b>	<b>Óptimo balance</b>
5	3,842.1	0.598	4,201.5	Sobre-segmentación
6	3,284.7	0.571	3,956.8	Pérdida de cohesión

#### Selección K=4:

- **Método del codo:** Inflexión clara en K=4
- **Máximo Silhouette Score:** 0.612 indica excelente separación
- **Interpretabilidad empresarial:** 4 segmentos permiten estrategias diferenciadas factibles

### 3.2.3 Configuración del Modelo Final

```
KMeans(  
    n_clusters=4,      # K óptimo determinado empíricamente  
    random_state=42,   # Reproducibilidad de resultados  
    n_init=20,         # 20 inicializaciones para estabilidad  
    max_iter=300       # Suficientes iteraciones para convergencia  
)
```

## 3.3 Criterios de Análisis del Modelo de Agrupación

### Criterio de Análisis 1: Caracterización Detallada de Clusters

Perfil de cada segmento identificado:

Cluster	Tamaño	% Total	Recency (días)	Frequency (compras)	Monetary (\$)	Interpretación
0	1,847	42.2%	89.3 ± 67.2	2.1 ± 1.8	\$347.85 ± \$298.67	<b>Clientes en Riesgo</b>
1	1,203	27.5%	35.7 ± 28.4	4.8 ± 2.1	\$789.23 ± \$456.12	<b>Clientes Leales</b>
2	892	20.4%	18.2 ± 15.6	8.7 ± 3.4	\$1,456.78 ± \$687.91	<b>Clientes VIP</b>
3	430	9.9%	52.8 ± 31.9	2.9 ± 1.5	\$2,347.91 ± \$1,247.83	<b>Grandes Compradores</b>

### Análisis interpretativo:

- **Cluster 0 (En Riesgo):** Mayor segmento, requiere estrategias de reactivación
- **Cluster 1 (Leales):** Balance entre frecuencia y valor, segmento core del negocio
- **Cluster 2 (VIP):** Máxima frecuencia, candidatos para programa de fidelización premium
- **Cluster 3 (Grandes Compradores):** Menor frecuencia pero máximo valor, estrategia de retención crítica

### Criterio de Análisis 2: Métricas de Calidad del Clustering

#### Evaluación cuantitativa de la segmentación:

Métrica	Valor	Rango	Interpretación
Silhouette Score	0.612	[-1, 1]	Excelente separación entre clusters
Calinski-Harabasz	4,347.1	[>0]	Alta cohesión intra-cluster y separación inter-cluster
Inercia (WCSS)	4,628.9	[>0]	Baja varianza dentro de clusters
Varianza PCA explicada	73.2%	[0%, 100%]	Los clusters capturan la mayoría de la variabilidad

### Análisis por cluster individual:

Cluster	Silhouette Score	Cohesión Interna	Interpretación de Calidad
0	$0.587 \pm 0.124$	Alta	Segmento bien definido, baja dispersión
1	$0.623 \pm 0.089$	Muy Alta	Segmento más cohesivo, características homogéneas
2	$0.641 \pm 0.098$	Muy Alta	Segmento premium claramente diferenciado
3	$0.578 \pm 0.156$	Moderada	Segmento especializado, mayor variabilidad interna

### Validación mediante visualización PCA:

- **Componente 1 (45.8% varianza):** Principalmente Monetary y Frequency
- **Componente 2 (27.4% varianza):** Principalmente Recency y características de producto
- **Separación visual clara** entre clusters en espacio bidimensional
- **Solapamiento mínimo** entre centroides de clusters

## 3.4 Interpretación Empresarial de los Clusters

### 3.4.1 Estrategias Diferenciadas por Segmento

#### Cluster 0 - Clientes en Riesgo (42.2%):

- **Estrategia:** Reactivación y recuperación
- **Tácticas:** Email marketing con descuentos, ofertas de productos complementarios
- **KPI objetivo:** Reducir Recency de 89 a 45 días
- **Budget recomendado:** 15% del presupuesto de marketing



#### Cluster 1 - Clientes Leales (27.5%):

- **Estrategia:** Retención y crecimiento
- **Tácticas:** Programa de puntos, cross-selling inteligente
- **KPI objetivo:** Incrementar Frequency de 4.8 a 6+ compras anuales
- **Budget recomendado:** 35% del presupuesto de marketing

#### Cluster 2 - Clientes VIP (20.4%):

- **Estrategia:** Maximización de valor y advocacy
- **Tácticas:** Servicio premium, acceso anticipado a productos, eventos exclusivos
- **KPI objetivo:** Mantener Frequency >8 e incrementar Monetary 20%
- **Budget recomendado:** 40% del presupuesto de marketing

#### Cluster 3 - Grandes Compradores (9.9%):

- **Estrategia:** Incremento de frecuencia
- **Tácticas:** Comunicación personalizada, recomendaciones basadas en histórico
- **KPI objetivo:** Incrementar Frequency de 2.9 a 4+ manteniendo Monetary
- **Budget recomendado:** 10% del presupuesto de marketing

### 3.4.2 Impacto de Negocio Proyectado

#### ROI estimado por implementación de segmentación:

- **Incremento en retención:** 15-25% según segmento
  - **Mejora en Customer Lifetime Value:** 18-30%
  - **Eficiencia en marketing spend:** 35-45% mejor targeting
  - **Incremento en cross-selling:** 20-40% más conversiones
-


## 4. REPOSITORIO Y ENTREGABLES TÉCNICOS

### 4.1 Estructura del Repositorio GitHub

URL del repositorio: <https://github.com/YiyoMb/extraccion-conocimiento-bd>

extraccion-conocimiento-bd/

- | — README.md # Documentación principal
- | — datos/
- | — Online\_Retail.xlsx # Dataset original
- | — notebooks/
- | — main.ipynb # Configuración inicial
- | — 01\_analisis\_exploratorio.ipynb # EDA completo
- | — 02\_modelo\_regresion.ipynb # Modelo de regresión
- | — 03\_modelo\_agrupacion.ipynb # Modelo de clustering
- | — 04\_modelo\_clasificacion.ipynb # Clasificación (Criterio DE)
- | — 05\_modelo\_asociacion.ipynb # Asociación (Criterio DE)
- | — 06\_dashboard\_interactivo.ipynb # Dashboard (Criterio AU)
- | — modelos/
- | — random\_forest\_regressor.pkl # Modelo de regresión entrenado
- | — linear\_regression.pkl # Baseline regresión
- | — scaler\_regression.pkl # Normalizador para regresión
- | — kmeans\_clustering.pkl # Modelo de clustering entrenado
- | — scaler\_clustering.pkl # Normalizador para clustering
- | — pca\_clustering.pkl # PCA para visualización
- | — customers\_with\_clusters.csv # Dataset con clusters asignados



```
| |— model_info_regression.json    # Metadatos modelo regresión
| |— clustering_info.json          # Metadatos modelo clustering
| |— visualizaciones/
| |— (gráficas generadas por los notebooks)
| |— documentos/
| |— criterio_sa_documento.pdf     # Este documento
```

## 4.2 Modelos Entregados - Regresión

Archivo principal: `random_forest_regressor.pkl`

- **Algoritmo:** RandomForestRegressor de scikit-learn
- **Características:** 15 variables de entrada procesadas
- **Rendimiento:**  $R^2 = 0.847$ , RMSE = 156.23
- **Tamaño:** 24.7 MB (100 árboles serializados)

Archivos complementarios:

- `scaler_regression.pkl`: StandardScaler para normalización de características
- `country_encoder.pkl`: LabelEncoder para codificación de países
- `model_info_regression.json`: Metadatos completos del modelo



### Instrucciones de uso:

```
import joblib

import pandas as pd

# Cargar modelo y componentes

modelo = joblib.load('modelos/random_forest_regressor.pkl')

scaler = joblib.load('modelos/scaler_regression.pkl')

# Preparar datos nuevos (mismo formato de entrenamiento)

nuevos_datos = pd.DataFrame(...) # 15 características

datos_escalados = scaler.transform(nuevos_datos)

# Realizar predicción

prediccion = modelo.predict(datos_escalados)
```

## 4.3 Modelos Entregados - Agrupación

**Archivo principal:** `kmeans_clustering.pkl`

- **Algoritmo:** KMeans de scikit-learn con K=4
- **Características:** 6 variables RFM + adicionales
- **Rendimiento:** Silhouette Score = 0.612
- **Centroides:** 4 perfiles de cliente bien diferenciados

### Archivos complementarios:

- `scaler_clustering.pkl`: StandardScaler para normalización RFM
- `pca_clustering.pkl`: PCA para visualización 2D
- `customers_with_clusters.csv`: Dataset con asignaciones de cluster

- `clustering_info.json`: Interpretaciones y metadatos

#### **Instrucciones de uso:**

```
import joblib
```

```
import pandas as pd
```

```
# Cargar modelo y componentes
```

```
clustering_model = joblib.load('modelos/kmeans_clustering.pkl')
```

```
scaler = joblib.load('modelos/scaler_clustering.pkl')
```

```
# Preparar datos RFM nuevos
```

```
rfm_data = calculate_rfm_features(transacciones_nuevas)
```

```
rfm_scaled = scaler.transform(rfm_data)
```

```
# Asignar clusters
```

```
clusters = clustering_model.predict(rfm_scaled)
```

---

## 5. CONCLUSIONES Y LECCIONES APRENDIDAS

### 5.1 Cumplimiento de Objetivos

#### ✓ Objetivos Técnicos Cumplidos:

- Implementación exitosa de Random Forest para regresión con  $R^2 = 0.847$
- Segmentación K-Means con excelente separación (Silhouette = 0.612)
- Análisis completo mediante múltiples criterios cuantitativos y cualitativos
- Entrega de modelos funcionales y reutilizables en repositorio

#### ✓ Objetivos de Aprendizaje Cumplidos:

- Comprensión profunda del pipeline completo de machine learning
- Experiencia práctica con preprocesamiento de datos reales "sucios"
- Dominio de métricas de evaluación y su interpretación empresarial
- Desarrollo de habilidades de comunicación técnica y documentación

### 5.2 Desafíos Superados

#### Desafío 1: Calidad de Datos

- **Problema:** 135,080 valores faltantes en CustomerID, transacciones negativas
- **Solución:** Pipeline de limpieza sistemático preservando 71% de datos originales
- **Aprendizaje:** La limpieza de datos consume 60-70% del tiempo pero es crítica

#### Desafío 2: Selección de Características

- **Problema:** Variables correlacionadas y escalas muy diferentes
- **Solución:** Ingeniería de características + análisis de importancia + normalización
- **Aprendizaje:** Las características derivadas a menudo superan a las originales

#### Desafío 3: Interpretabilidad vs Rendimiento

- **Problema:** Trade-off entre modelos simples interpretables y complejos precisos
- **Solución:** Implementar ambos y usar feature importance para explicabilidad
- **Aprendizaje:** Los stakeholders valoran tanto la precisión como la explicabilidad



## 5.3 Impacto y Aplicabilidad

### Valor Empresarial Generado:

- **Predicción de ingresos:** Modelo permite estimaciones con 84.7% de precisión
- **Segmentación automatizada:** Reemplaza análisis manual subjetivo
- **Personalización:** Base para estrategias de marketing diferenciadas
- **Eficiencia operativa:** Automatización de procesos de análisis de clientes

### Escalabilidad del Enfoque:

- Metodología aplicable a otros datasets de e-commerce
- Pipeline reutilizable para análisis periódicos automáticos
- Framework extensible para incorporar nuevas características

## 5.4 Próximos Pasos y Mejoras

### Mejoras Técnicas Identificadas:

1. **Hyperparameter tuning:** Grid search para optimización de parámetros
2. **Feature engineering avanzado:** Características temporales más sofisticadas
3. **Ensemble methods:** Combinar múltiples algoritmos para mejor rendimiento
4. **Real-time scoring:** API para predicciones en tiempo real

### Extensiones del Análisis:

1. **Análisis de churn:** Predecir qué clientes abandonarán
  2. **Lifetime value:** Estimar valor total del cliente
  3. **Recommender systems:** Recomendaciones personalizadas de productos
  4. **Análisis de canasta:** Productos que se compran juntos
-



## 6. REFERENCIAS TÉCNICAS

### 6.1 Fuentes de Datos

- **UCI Machine Learning Repository** - Online Retail Dataset
- **Dua, D. and Graff, C.** (2019). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.

### 6.2 Librerías y Herramientas Utilizadas

- **Python 3.11+** - Lenguaje de programación principal
- **pandas 2.2.2** - Manipulación y análisis de datos
- **scikit-learn 1.6.1** - Algoritmos de machine learning
- **matplotlib 3.10.0** - Visualización de datos
- **seaborn** - Visualización estadística avanzada
- **numpy 2.0.2** - Computación numérica
- **Google Colab** - Entorno de desarrollo
- **GitHub** - Control de versiones y repositorio

### 6.3 Metodologías Aplicadas

- **CRISP-DM** - Metodología para proyectos de data mining
  - **RFM Analysis** - Framework de segmentación de clientes
  - **Cross-validation** - Validación de modelos de machine learning
  - **Principios de Clean Code** - Desarrollo de código mantenible
-

## ANEXOS

### Anexo A: Código de Configuración del Entorno

*# Instalación de dependencias*

```
!pip install plotly dash mlxtend wordcloud
```

*# Configuración de visualizaciones*

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
plt.style.use('seaborn-v0_8')
```

```
sns.set_palette("husl")
```

```
plt.rcParams['figure.figsize'] = (12, 8)
```

### Anexo B: Estadísticas Descriptivas del Dataset Final

- **Dimensiones:** 387,284 transacciones × 15 características
- **Período temporal:** 2010-12-01 a 2011-12-09
- **Países únicos:** 38 países
- **Clientes únicos:** 4,372 clientes
- **Productos únicos:** 3,684 productos

### Anexo C: Configuración de Hiperparámetros


*# Random Forest Regressor*

```
rf_params = {
```

```
    'n_estimators': 100,
```

```
    'max_depth': 15,
```

```
    'min_samples_split': 5,
```



```
'min_samples_leaf': 2,  
'random_state': 42  
}
```

*# K-Means Clustering*

```
kmeans_params = {  
    'n_clusters': 4,  
    'random_state': 42,  
    'n_init': 20,  
    'max_iter': 300  
}
```

---

**Contacto:** 2022371075@uteq.edu.mx

**Repositorio:** <https://github.com/YiyoMb/extraccion-conocimiento-bd>

**Última actualización:** 31 de Julio de 2025