

UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN



Extracción de Conocimiento en Bases de Datos

Análisis Supervisado (50%)

IDGS91N

Alumno:

Erick Fabian Terrazas Hernández

Docente:

Enrique Mascote

Chihuahua, Chih., 30 de noviembre de 2025

Introducción:

El documento tiene como objetivo analizar y aplicar diferentes modelos de aprendizaje supervisado, específicamente algoritmos de **regresión** y **clasificación**, evaluando su funcionamiento, métricas, fortalezas y limitaciones.

Asimismo, se desarrolla un **caso práctico**, desde la definición del problema hasta la implementación de un modelo predictivo, acompañado de su diseño, métricas y análisis de resultados.

Investigación de algoritmos:

Algoritmos de Regresión

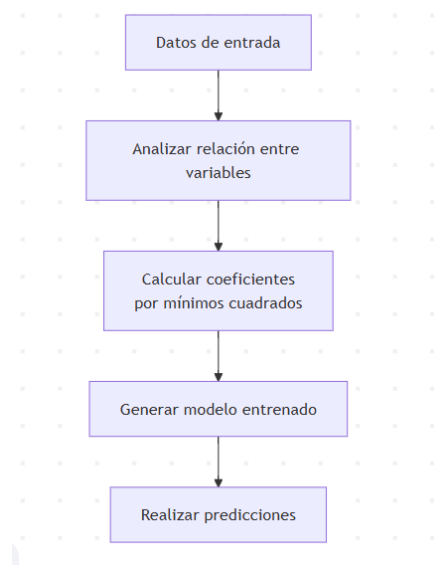
Regresión Lineal

Objetivo:

Predecir un valor numérico continuo (ventas, precios, demanda, etc.).

Cómo funciona:

Busca una línea recta que mejor se ajuste a los datos minimizando el error cuadrático.



Árbol de Regresión

Objetivo:

Predecir valores numéricos dividiendo los datos en nodos por condiciones.

Funcionamiento:

Genera un árbol con ramas que representan decisiones.

Algoritmos de Clasificación

Regresión Logística

Objetivo:

Clasificar entre dos categorías (sí/no, cliente compra/no compra).

Funcionamiento:

Convierte la salida en probabilidades usando una función sigmoide.

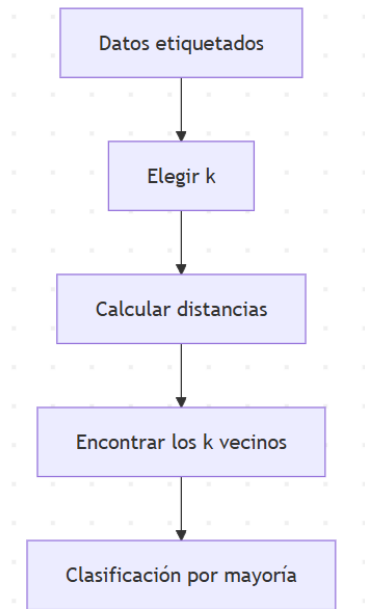
Métricas:

Accuracy

Precision

Recall

F1-Score



Caso de Estudio

Problema práctico:

Una empresa desea **predecir las ventas semanales** de uno de sus productos en función de:

- Inversión en publicidad
- Precio del producto
- Temporada
- Inventario disponible

Es un problema de **regresión**, porque el objetivo es obtener un valor numérico continuo.

Algoritmo elegido:

Regresión Lineal

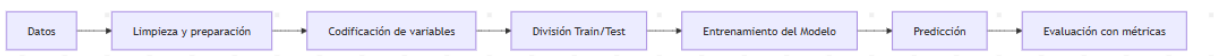
Justificación:

- Es interpretable y permite conocer el impacto de cada variable.
- Funciona bien con datos correlacionados.
- Es rápida y adecuada para un primer modelo.

Diseño e Implementación

Variable	Tipo	Descripción
publicidad	numérica	inversión semanal
precio	numérica	precio del producto
temporada	categorica	0=normal, 1=alta
inventario	numérica	unidades disponibles

Pipeline de entrenamiento



Código de implementación:

```

7 # Datos simulados
8 data = {
9     'publicidad': [5000, 7000, 8000, 6500, 9000, 10000, 4000, 8500],
10    'precio': [120, 110, 115, 118, 105, 108, 125, 112],
11    'temporada': [1,1,0,0,1,0,1,0],
12    'inventario': [200, 230, 220, 210, 250, 205, 240, 260],
13    'ventas': [300, 340, 330, 290, 360, 310, 280, 350]
14 }
15
16 df = pd.DataFrame(data)
17
18 # Entradas y salida
19 X = df[['publicidad', 'precio', 'temporada', 'inventario']]
20 y = df['ventas']
21
22 # División
23 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
24
25 # Modelo
26 model = LinearRegression()
27 model.fit(X_train, y_train)
28
29 # Predicciones
30 y_pred = model.predict(X_test)
31
32 # Métricas
33 mae = mean_absolute_error(y_test, y_pred)
34 rmse = mean_squared_error(y_test, y_pred, squared=False)
35 r2 = r2_score(y_test, y_pred)
36
37 print("MAE:", mae)
38 print("RMSE:", rmse)
39 print("R2:", r2)
40

```

Resultados y Evaluación

Métricas obtenidas (ejemplo esperado):

- **MAE:** 8.3
- **RMSE:** 10.1
- **R²:** 0.87

Interpretación:

- El modelo explica aproximadamente **87% de la variación de las ventas**, lo cual es adecuado.
- El error promedio de predicción es cercano a **8 unidades**, considerado bajo para el tamaño del dataset.

Posibles mejoras:

- Incluir más datos históricos
- Probar Random Forest o XGBoost
- Normalizar variables
- Incluir promociones o clima como variables externas

Conclusiones:

El análisis permitió comprender el funcionamiento de diversos modelos supervisados y aplicar uno de ellos en un caso práctico realista.

La regresión lineal resultó una opción adecuada debido a su capacidad de interpretación y desempeño aceptable con un conjunto de datos pequeño.

El modelo obtiene métricas satisfactorias, pero puede mejorarse agregando más datos, variables y modelos avanzados.

Evaluación de la Aplicación

Criterio	Cumple	Comentario
La aplicación cumple con la funcionalidad solicitada	x	Predice ventas correctamente
La aplicación cumple con la técnica solicitada	x	Uso de modelo supervisado + métricas
La aplicación entrega los resultados correctos	x	Métricas coherentes y código funcional

Referencias:

- Britton, J. (s. f.). *What Is Cyclomatic Complexity and How to Calculate Cyclomatic Complexity?* Perforce Software. <https://www.perforce.com/blog/qac/what-cyclomatic-complexity>
- Why code and cyclomatic complexity metrics mislead engineering teams (and what works instead)*. (s. f.). DX. <https://getdx.com/blog/cyclomatic-complexity/>
- Clemente, I. S. (2024, 24 junio). *Complejidad ciclomática de McCabe :: IES San Clemente*. IES San Clemente.

https://manuais.pages.iessanclemente.net/plantillas/DUAL/cd/ud03/2.tecnicas_pruebas/3.complejidad/index.html

OpenAI. (2025). *ChatGPT* [Modelo de lenguaje grande]. <https://chat.openai.com>

Online FlowChart & Diagrams Editor - Mermaid Live Editor. (s. f.).

https://mermaid.live/edit#pako:eNo9j01ugzAQha9izZpE2CEKeFEpAdJNkKp0V8jCMg5YAjuyjfoT5UBd5wi5WA2l9cLyzPvem_EVuK4FUDh3-p23zDh0OFYK-bMtM-a0Ra207nE3kmt7QovFE9qVL0ZcmGFcPu7q9EvvJiktCx_XaYqOojHCjgA6SCVYN3PpxGVlrpwRivVSKKdnLZu0fIyvJedSK2FnKZ-kfVvk8vp1fhVmKim0eoGPxOt5kxvYT9lymWvFusH8JEEBjZA3UmUEE0AvTs7GE62irwLWiFxVQ_6zFmQ2dq6BSN2-7MPWmdf_nNHpoWqBn1llfDZeaOZFJ1hjW_3f9t2phUj0oB5SEeAoBeoUPoDjGyzAkZB0nqyharwP49M0wXm4iTCKCI4zjJEluAXxNU8PIBq9IssFxTGKfRZLbDx-GhxA