UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

EXTRACCION DE CONOCIMIENTO EN BASES DE DATOS

III.1. ANÁLISIS SUPERVISADO

IDGS91N

PRESENTA:

SEBASTIÁN ACOSTA ORTIZ

DOCENTE:

LUIS ENRIQUE MASCOTE CANO

Chihuahua, Chih., 30 de noviembre de 2025

Introducción

El análisis supervisado es un conjunto de técnicas de aprendizaje automático cuyo objetivo es predecir una variable objetivo a partir de datos previamente etiquetados. Este proceso incluye la selección del algoritmo adecuado, la preparación de datos, el entrenamiento, la evaluación y la interpretación de los resultados. El presente trabajo tiene como propósito investigar algoritmos de regresión y clasificación, analizar sus características, aplicarlos a un caso práctico y evaluar su desempeño mediante métricas estandarizadas.

Investigación de algoritmos

Algoritmos de Regresión

A) Regresión Lineal

Objetivo:

Predecir valores numéricos continuos mediante una relación lineal entre variables.

Principio de funcionamiento:

Ajusta una ecuación del tipo

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

Minimiza el error cuadrático entre predicciones y valores reales (mínimos cuadrados).

Métricas típicas:

- MAE (Error absoluto medio)
- MSE / RMSE (Error cuadrático medio / raíz cuadrática)
- R^2 (coeficiente de determinación)

Fortalezas:

- Simple, rápido y fácil de interpretar.

- Útil como línea base.

Limitaciones:

- No captura relaciones no lineales.
- Sensible a valores atípicos.

B) Árboles de Decisión para Regresión

Objetivo:

Predecir valores numéricos dividiendo el espacio de datos mediante reglas jerárquicas.

Principio de funcionamiento:

Crea nodos que dividen los datos con base en reducir la varianza. Cada hoja contiene un valor promedio.

Métricas típicas:

- MAE
- RMSE
- R^2

Fortalezas:

- Captura relaciones no lineales.
- Fácil interpretación.

Limitaciones:

- Puede sobre ajustarse.
- Sensible a pequeñas variaciones en los datos si no se poda.

Algoritmos de Clasificación

A) K-Nearest Neighbors (KNN)

Objetivo:

Clasificar datos según los "k" vecinos más cercanos.

Funcionamiento:

- Calcula la distancia (euclidiana usualmente) entre un punto nuevo y los existentes.
- La clase mayoritaria de los vecinos determina la clasificación.

Métricas típicas:

- Accuracy
- Precision / Recall
- F1-score

Fortalezas:

- Simple y no requiere entrenamiento.
- Efectivo con fronteras no lineales.

Limitaciones:

- Lento con grandes volúmenes.
- Sensible a escalado de datos.

B) Regresión Logística

Objetivo:

Clasificar una instancia en categorías binarias (0/1).

Funcionamiento:

Modela la probabilidad usando la función sigmoide:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots)}}$$

Métricas típicas:

- Accuracy
- Precision
- Recall
- AUC–ROC

Fortalezas:

- Interpretable.
- Rápida y robusta.

Limitaciones:

- Útil solo para fronteras lineales.
- No escala bien para datos muy complejos sin regularización.

Caso de estudio y justificación del algoritmo

Caso práctico: Clasificación de clientes con riesgo de abandono

Una empresa digital desea predecir si un cliente abandonará su servicio en los próximos meses. El dataset contiene:

- *Edad*
- *Meses de permanencia*
- *Uso mensual del servicio*
- *Tickets de soporte abiertos*
- *Etiqueta*:
    - 1 = riesgo de abandono
    - 0 = cliente estable

Algoritmo elegido: Regresión Logística

Justificación:

- El problema es binario, por lo que la regresión logística es adecuada.
- Genera probabilidades interpretables.
- Permite identificar variables con mayor influencia.
- Es rápida y eficaz como primera aproximación antes de modelos más complejos como Random Forest o KNN.

Diseño e implementación

Variables y estructura de datos

Variables independientes (X):

- edad
- meses_permanencia
- uso_mensual
- tickets_soporte

Variable dependiente (y):

- abandono (0/1)

Pipeline de entrenamiento

1. Cargar datos
2. Separar entrenamiento/prueba
3. Escalar variables
4. Entrenar modelo

5. Evaluar métricas (accuracy, recall, F1)

```
'import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score

# Dataset simulado
data = {
    "edad": [25,40,32,19,50,45,22,37,29,55],
    "meses": [3,18,10,1,24,20,5,12,8,30],
    "uso": [12,3,8,15,4,2,14,6,10,1],
    "tickets": [3,0,1,4,1,0,2,1,3,0],
    "abandono": [1,0,0,1,0,0,1,0,1,0]
}

df = pd.DataFrame(data)

# Variables
X = df[["edad","meses","uso","tickets"]]
y = df["abandono"]

# División
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Escalado
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Modelo
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Predicciones
y_pred = model.predict(X_test_scaled)

# Métricas
acc = accuracy_score(y_test, y_pred)
```

```python
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", acc)
print("Precision:", prec)
print("Recall:", rec)
print("F1-score:", f1)
```

Resultados y evaluación

Tras entrenar el modelo de regresión logística con el dataset simulado, se obtuvieron los siguientes valores de desempeño:

```
PS C:\Users\Sebastiana\Documents\Uni\9no Tetra\ECBDEM\U3> & C:/Us
Uni/9no Tetra/ECBDEM/U3/codes/ev1u3.py"
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
```

Análisis

- El modelo obtuvo un rendimiento **perfecto (100%)** en el conjunto de prueba.
- Esto ocurre porque:
    1. El dataset es **muy pequeño (10 ejemplos)**.
    2. Los patrones entre variables y la etiqueta están **muy limpios y separados**.
    3. No existe ruido significativo que dificulte la predicción.

¿Es esto realista?

No necesariamente. En problemas reales, obtener métricas tan perfectas suele ser una señal de **sobreajuste**, o de que el conjunto de prueba no representa suficiente variedad.

Posibles mejoras y siguientes pasos

- Incrementar el número de datos para evaluar generalización real.
- Introducir ruido o casos frontera para hacer la evaluación más robusta.
- Comparar con otros algoritmos (KNN, Random Forest).
- Realizar validación cruzada k-fold.

Referencias

AI-FutureSchool. (2025, noviembre 30). AI-FutureSchool – Algoritmos de regresión y su aplicación en datos. Recuperado el 30 de noviembre de 2025, de AI-FutureSchool website: https://www.ai-futureschool.com/es/informatica/algoritmos-de-regresion-guia-completa.php

Algoritmos de regresión. (2022, agosto 18). Recuperado el 30 de noviembre de 2025, de Datapeaker website: https://datapeaker.com/big-data/algoritmos-de-regresion-5-algoritmos-de-regresion-que-debes-conocer/