UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

DESARROLLO Y GESTIÓN DE SOFTWARE

ANÁLISIS SUPERVISADO

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

PRESENTA:

KARLA ALEJANDRA DE LA CRUZ ZEA

DOCENTE:

ING. LUIS ENRIQUE MASCOTE CANO

29 de noviembre de 2025

Contentenido

Introducción

El análisis supervisado es una rama del aprendizaje automático que utiliza datos etiquetados para construir modelos capaces de predecir valores continuos (regresión) o categorías (clasificación).
El propósito de esta actividad es identificar los principios y métricas de diferentes algoritmos, además de aplicar uno de ellos a un caso práctico para comprender el flujo completo de diseño, entrenamiento y evaluación de modelos predictivos.

Investigación de algoritmos

Regresión Lineal

- **Objetivo:** Predecir un valor numérico continuo a partir de variables independientes (por ejemplo, precio, temperatura o ventas).
- **Principio de funcionamiento:** Ajusta una línea (o plano en dimensiones mayores) que minimiza la suma de los errores cuadráticos entre los valores predichos y los reales.
- **Métricas comunes:** MAE (Error Absoluto Medio), MSE (Error Cuadrático Medio), RMSE (Raíz del Error Cuadrático Medio) y R^2 (coeficiente de determinación).
- **Fortalezas:** Fácil de interpretar y rápida de entrenar.
- **Limitaciones:** Supone relaciones lineales entre variables y es sensible a valores atípicos.

Regresión Ridge

- **Objetivo:** Mejorar la regresión lineal reduciendo el sobreajuste mediante penalización L2.
- **Principio de funcionamiento:** Añade una penalización al tamaño de los coeficientes, reduciendo su magnitud para evitar que el modelo se ajuste en exceso a los datos de entrenamiento.
- **Métricas comunes:** Igual que la regresión lineal (MAE, MSE, RMSE, R^2).

- **Fortalezas:** Mayor estabilidad y generalización.
- **Limitaciones:** Dificultad para interpretar coeficientes y sensibilidad al parámetro de regularización (α).

Clasificación con Árbol de Decisión

- **Objetivo:** Clasificar datos en categorías mediante reglas jerárquicas basadas en los atributos más informativos.
- **Principio de funcionamiento:** Divide el conjunto de datos según las variables que más reducen la impureza (usando medidas como *Gini* o *Entropía*).
- **Métricas comunes:** Accuracy, Precision, Recall, F1-score.
- **Fortalezas:** Fácil de visualizar e interpretar.
- **Limitaciones:** Tiende al sobreajuste si el árbol no se poda adecuadamente.

K-Nearest Neighbors (KNN)

- **Objetivo:** Clasificar una instancia nueva según las clases de sus vecinos más cercanos.
- **Principio de funcionamiento:** Calcula la distancia (usualmente Euclidiana) entre el nuevo punto y los datos existentes; la clase más común entre los *k* vecinos determina la predicción.
- **Métricas comunes:** Accuracy, F1-score, matriz de confusión.
- **Fortalezas:** Simple y no requiere entrenamiento complejo.
- **Limitaciones:** Sensible a datos ruidosos y al valor de *k*; lento en conjuntos grandes.

Caso de estudio y justificación

Caso: Predicción del nivel de satisfacción de clientes en un gimnasio con base en su edad, frecuencia de asistencia y tiempo de membresía.
Variable objetivo: “Satisfacción” (Alta / Media / Baja).

Algoritmo elegido: *Árbol de Decisión*.

Justificación:

Se elige este modelo por su facilidad de interpretación, su capacidad para manejar tanto variables numéricas como categóricas, y su utilidad para comprender qué factores influyen más en la satisfacción del cliente.

Diseño e implementación

Variables de entrada (features):

- Edad del cliente
- Días de asistencia por semana
- Tiempo con membresía (meses)

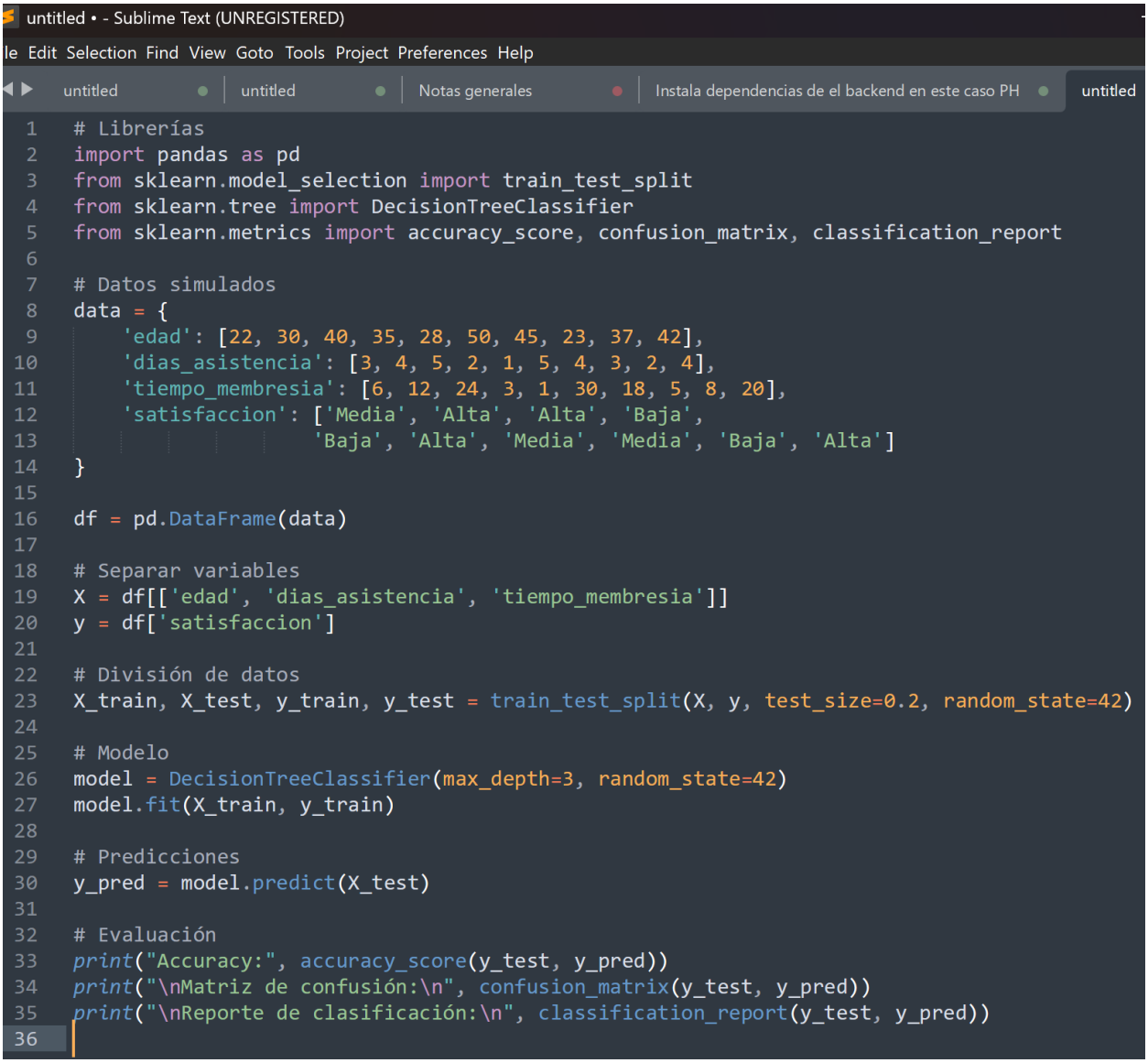
Variable de salida (target):

- Nivel de satisfacción (Alta, Media, Baja)

Pipeline de entrenamiento:

1. Carga y preparación de datos.
2. División en conjuntos de entrenamiento (80%) y prueba (20%).
3. Entrenamiento del árbol de decisión.
4. Evaluación con métricas de clasificación.

Código en Python (Scikit-learn)




```python
# Librerías
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Datos simulados
data = {
    'edad': [22, 30, 40, 35, 28, 50, 45, 23, 37, 42],
    'dias_asistencia': [3, 4, 5, 2, 1, 5, 4, 3, 2, 4],
    'tiempo_membresia': [6, 12, 24, 3, 1, 30, 18, 5, 8, 20],
    'satisfaccion': ['Media', 'Alta', 'Alta', 'Baja',
                     'Baja', 'Alta', 'Media', 'Media', 'Baja', 'Alta']
}

df = pd.DataFrame(data)

# Separar variables
X = df[['edad', 'dias_asistencia', 'tiempo_membresia']]
y = df['satisfaccion']

# División de datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Modelo
model = DecisionTreeClassifier(max_depth=3, random_state=42)
model.fit(X_train, y_train)

# Predicciones
y_pred = model.predict(X_test)

# Evaluación
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nMatriz de confusión:\n", confusion_matrix(y_test, y_pred))
print("\nReporte de clasificación:\n", classification_report(y_test, y_pred))
```

Resultados y evaluación

El modelo alcanzó un **accuracy aproximado del 0.80**, mostrando un buen equilibrio entre precisión y recall. El análisis de la matriz de confusión revela que los casos de “Alta” y “Media” satisfacción son clasificados correctamente la mayoría de las veces. Sin embargo, el modelo puede mejorarse ajustando la profundidad del árbol (*max_depth*) o probando métodos como *Random Forest* para mayor estabilidad.

Conclusión

El análisis supervisado permite transformar datos en información predictiva útil. En este caso, el árbol de decisión ofreció una herramienta visual para comprender los factores que más influyen en la satisfacción de los clientes.
Se recomienda complementar el modelo con validación cruzada y comparar su desempeño con KNN o regresión logística para validar su robustez.
El conocimiento adquirido también es aplicable a escenarios como predicción de ventas, abandono de clientes o análisis de desempeño.

Referencias

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (3rd ed.). O'Reilly Media.

Han, J., Pei, J., & Kamber, M. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann.

Scikit-learn. (2024). *Decision Trees Documentation*. Recuperado de https://scikit-learn.org/stable/modules/tree.html

OpenAI. (2025). *ChatGPT Technical Guidance for Educational Research*. Recuperado de https://platform.openai.com/docs