EXTRACCIÓN DE CONOCIMIENTOS EN BASES DE DATOS

ING. LUIS ENRIQUE MASCOTE CANO

Universidad Tecnológica de Chihuahua

ANÁLISIS

SUPERVISADO

Lic. Ricardo Hernández Martínez

Fecha de Entrega: 30/11/25

Introducción

El aprendizaje automático (Machine Learning) ofrece un conjunto de algoritmos capaces de resolver problemas de predicción y clasificación mediante el análisis de datos. En este documento se investigan cuatro algoritmos ampliamente utilizados —dos de regresión y dos de clasificación— describiendo su objetivo, funcionamiento, métricas comunes y limitaciones. Posteriormente, se desarrolla un caso de estudio donde se selecciona uno de estos algoritmos para resolver un problema aplicado, incluyendo diseño del modelo, justificación y código de implementación.

Sección 1: Investigación de Algoritmos

1. Algoritmos de Regresión

1.1 Regresión Lineal

Objetivo: Predecir un valor numérico continuo mediante una combinación lineal de las variables independientes.

Principio de funcionamiento: Calcula los coeficientes que minimizan el error cuadrático entre las predicciones y los valores reales. El aprendizaje se basa en encontrar la línea (o hiperplano) que mejor ajusta los datos.

Métricas típicas: MAE, MSE, RMSE, R^2.

Fortalezas: Sencilla de interpretar, rápida de entrenar, adecuada cuando existe relación lineal.

Limitaciones: No captura relaciones no lineales; sensible a valores atípicos.

1.2 Random Forest Regressor

Objetivo: Predecir valores numéricos mediante un conjunto de árboles de decisión.

Principio de funcionamiento: Construye múltiples árboles de decisión sobre subconjuntos de datos y promedia sus predicciones para mejorar la estabilidad y precisión.

Métricas típicas: MAE, MSE, RMSE, R^2.

Fortalezas: Captura patrones no lineales, robusto a valores atípicos, evita sobreajuste mejor que un solo árbol.

Limitaciones: Menos interpretable, mayor costo computacional.

Algoritmos de Clasificación

2.1 K-Nearest Neighbors (KNN)

Objetivo: Clasificar una instancia asignándole la clase mayoritaria entre sus vecinos más cercanos.

Principio de funcionamiento: Calcula la distancia entre la instancia a predecir y las observaciones del conjunto de entrenamiento; selecciona los *k* más cercanos.

Métricas típicas: Accuracy, precisión, recall, F1-score.

Fortalezas: Fácil de implementar, no requiere entrenamiento explícito.

Limitaciones: Costoso en predicción, sensible a la escala de los datos y al ruido.

2.2 Árboles de Decisión

Objetivo: Clasificar datos mediante reglas secuenciales basadas en las características.

Principio de funcionamiento: Divide los datos de forma recursiva según el valor que aporta mayor ganancia de información.

Métricas típicas: Accuracy, precisión, recall, F1-score.

Fortalezas: Interpretabilidad, manejo de datos numéricos y categóricos.

Limitaciones: Tienden al sobreajuste si no se regulan.

Sección 2: Solución del Caso de Estudio

Caso práctico

Se desea predecir las ventas semanales de una tienda minorista utilizando variables como inversión en publicidad, días festivos y tráfico de clientes.

Justificación del algoritmo elegido

Se selecciona **Random Forest Regressor**, debido a su capacidad para manejar relaciones no lineales y múltiples variables con diferentes distribuciones. Además, reduce el riesgo de sobreajuste presente en un árbol individual.

Diseño del modelo

- **Variables de entrada:** inversión en marketing, visitas a la tienda, día de la semana, indicador de fecha especial.
- **Estructura de datos:** DataFrame con columnas numéricas y categóricas codificadas.
- **Pipeline:**
    1. Carga y limpieza de datos.
    2. Codificación de variables categóricas.
    3. Escalado opcional.
    4. División en entrenamiento y prueba.
    5. Entrenamiento del modelo.
    6. Evaluación.

Implementación (Python + scikit-learn)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Datos simulados
import numpy as np
```

```
np.random.seed(0)

data = pd.DataFrame({
'marketing': np.random.randint(1000, 5000, 200),
'visitas': np.random.randint(300, 2000, 200),
'festivo': np.random.randint(0, 2, 200),
'ventas': np.random.randint(5000, 20000, 200)
})

X = data[['marketing', 'visitas', 'festivo']]
y = data['ventas']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

model = RandomForestRegressor(n_estimators=100, random_state=0)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)

mae, rmse, r2
```

Análisis de resultados

El modelo suele ofrecer valores competitivos de MAE y RMSE gracias al uso de múltiples árboles. Un R^2 cercano a 1 indica buena capacidad predictiva. Sin embargo, el rendimiento puede mejorar mediante:

- ajuste de hiperparámetros (número de árboles, profundidad máxima),
- ingeniería de características (nuevas variables relevantes),
- eliminación de ruido en los datos.

Conclusiones

Los algoritmos de regresión y clasificación ofrecen enfoques distintos para resolver problemas predictivos. Random Forest destaca por su robustez y capacidad de generalización, lo que lo hace adecuado para el caso de predicción de ventas. El análisis desarrollado demuestra cómo la selección correcta del algoritmo, acompañada de un diseño adecuado del modelo, permite obtener resultados confiables.

Referencias

Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5–32.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd ed.). Springer.

Pedregosa, F., et al. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.