UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Desarrollo de software

Extracción de Conocimiento en Bases de Datos

III.1. Análisis Supervisado (50%)

IDGS91N

PRESENTA:

Juan Carlos Medina Sánchez

DOCENTE:

Enrique Mascote

Chihuahua, Chih., 29 de noviembre de 2025

1. Introducción

El aprendizaje supervisado es una rama del aprendizaje automático en la que se entrena un modelo a partir de ejemplos etiquetados. Es decir, para cada observación se conocen tanto las variables de entrada (features) como la salida deseada (label). El objetivo del modelo es aprender una función que relacione ambas partes y que pueda generalizar a datos nuevos.

Dentro del aprendizaje supervisado existen dos tareas principales: **regresión**, donde se predicen valores numéricos continuos (por ejemplo, ventas, temperatura, precio), y **clasificación**, donde se predicen categorías (por ejemplo, "compra/no compra", "fraude/no fraude", "aprobado/reprobado").

En este documento se presentan dos algoritmos de regresión y dos de clasificación, describiendo su objetivo, principio de funcionamiento, métricas de evaluación, fortalezas y limitaciones. Posteriormente, se desarrolla un caso práctico de predicción de ventas, se justifica la elección de un algoritmo, se diseña el modelo y se muestra una implementación básica en Python utilizando la librería scikit-learn. Finalmente, se analizan los resultados y se plantean posibles mejoras.

2. Investigación de algoritmos

2.1. Regresión lineal (Regresión)

Qué resuelve (objetivo)
La regresión lineal busca modelar la relación entre una variable dependiente continua yy una o varias variables independientes X, mediante una combinación lineal de estas últimas. Se usa para **predecir valores numéricos** (por ejemplo, ventas, ingresos, precio de casas).

Principio de funcionamiento (proceso)
La regresión lineal ajusta una recta (o hiperplano) que minimiza el error cuadrático entre las predicciones y los valores reales. El modelo tiene la forma:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Los coeficientes $\beta_i$se estiman típicamente por **mínimos cuadrados ordinarios (OLS)**, minimizando la suma de los errores al cuadrado.

Métricas de evaluación típicas

- **MAE (Mean Absolute Error)**
- **MSE (Mean Squared Error)**
- **RMSE (Root Mean Squared Error)**

- R^2**(Coeficiente de determinación)**

Fortalezas

- Modelo simple, rápido y fácil de interpretar.
- Permite entender el efecto (signo y magnitud) de cada variable.
- Buena base como modelo de referencia (baseline).

Limitaciones

- Supone relación lineal entre variables.
- Sensible a outliers.
- Requiere condiciones estadísticas (homocedasticidad, independencia de errores, etc.) que no siempre se cumplen.

2.2. Bosques aleatorios para regresión (Random Forest Regressor)

Qué resuelve (objetivo)
Un bosque aleatorio de regresión predice variables continuas combinando muchos árboles de decisión. Es útil cuando la relación entre variables es compleja y no lineal.

Principio de funcionamiento (proceso)

- Construye muchos **árboles de decisión** sobre diferentes subconjuntos de datos y características (bootstrap + selección aleatoria de features).
- Cada árbol produce una predicción.
- La predicción final del bosque es la **media** de las predicciones de todos los árboles.
- El uso de muchos árboles reduce el sobreajuste típico de un árbol de decisión individual.

Métricas de evaluación típicas

- MAE
- MSE / RMSE
- R^2

Fortalezas

- Capta relaciones **no lineales** y **interacciones** entre variables.
- Menos sensible al ruido que un árbol individual.

- Maneja bien grandes conjuntos de datos y muchas variables.
- Proporciona importancia de características (feature importance).

Limitaciones

- Menos interpretable que un modelo lineal simple.
- Entrenamiento más costoso computacionalmente.
- Requiere ajuste de hiperparámetros (número de árboles, profundidad, etc.).

2.3. Regresión logística (Clasificación)

Qué resuelve (objetivo)
La regresión logística se utiliza para **clasificación binaria**, es decir, cuando la salida tiene dos clases (por ejemplo, "cliente que abandona" vs "cliente que permanece"). Predice la probabilidad de pertenecer a una clase.

Principio de funcionamiento (proceso)

- Modela la probabilidad de que $y = 1$dado Xmediante la **función logística**:

$$P(y = 1 \mid X) = \sigma(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p),$$

donde $\sigma(z) = \frac{1}{1+e^{-z}}$.

- Los parámetros βse estiman mediante **máxima verosimilitud**.
- Se aplica un umbral (por ejemplo, 0.5) sobre la probabilidad para decidir la clase.

Métricas de evaluación típicas

- Accuracy (exactitud)
- Precision, Recall (sensibilidad), F1-Score
- AUC-ROC (Área bajo la curva ROC)

Fortalezas

- Modelo relativamente interpretable.
- Probabilístico: entrega probabilidades, no solo clases.
- Funciona bien como baseline en muchos problemas de clasificación.

Limitaciones

- Supone una frontera de decisión aproximadamente lineal.
- Puede tener problemas cuando las clases no son linealmente separables.
- Sensible a la multicolinealidad y a outliers.

2.4. Bosques aleatorios para clasificación (Random Forest Classifier)

Qué resuelve (objetivo)
Random Forest para clasificación resuelve problemas de **clasificación binaria o multiclase**, combinando múltiples árboles de decisión.

Principio de funcionamiento (proceso)

- Igual que en regresión, entrena muchos árboles de decisión sobre diferentes subconjuntos de datos y características.
- Cada árbol vota por una clase.
- La predicción final es la **clase mayoritaria** entre los árboles (votación por mayoría).

Métricas de evaluación típicas

- Accuracy
- Precision, Recall, F1-Score
- Matriz de confusión
- AUC-ROC (para problemas binarios)

Fortalezas

- Excelente desempeño en muchos problemas prácticos sin mucho preprocesamiento.
- Captura relaciones no lineales y complejas.
- Menos propenso a sobreajuste que un árbol individual.

Limitaciones

- Menos interpretable que modelos lineales o un solo árbol.
- Consume más memoria y tiempo de cómputo.
- Puede requerir ajuste de muchos hiperparámetros.

3. Caso de estudio y justificación

3.1. Descripción del problema

Se plantea un caso práctico de **predicción de ventas mensuales** para una tienda minorista (por ejemplo, una tienda en línea). El objetivo es estimar las ventas del próximo mes a partir de información histórica.

Variables disponibles por mes:

- Inversión en marketing digital (MXN).
- Número de visitas al sitio web.
- Número de promociones/ descuentos activos.
- Precio promedio de los productos.
- Estación del año (codificada, por ejemplo, mediante variables dummy).
- Ventas totales del mes anterior (target a predecir para el siguiente mes o usar ventas del mes actual como target).

El objetivo es construir un modelo que permita **predecir las ventas mensuales futuras** para apoyar decisiones de inventario y marketing.

3.2. Justificación del algoritmo elegido

De los algoritmos de la sección 2, para este problema se elige **Bosques Aleatorios para regresión (Random Forest Regressor)** por las siguientes razones:

- La relación entre las variables explicativas (marketing, visitas, promociones, precio) y las ventas **no es necesariamente lineal**.
- Random Forest puede capturar **interacciones complejas** (por ejemplo, el efecto combinado de marketing y descuentos).
- Es robusto frente a outliers y al ruido en los datos.
- No requiere una gran cantidad de supuestos estadísticos sobre la distribución de los datos.

La regresión lineal se podría utilizar como modelo base, pero se espera que Random Forest obtenga mejor desempeño al modelar relaciones no lineales.

4. Diseño e implementación

4.1. Variables de entrada y estructura de datos

La estructura de datos es una tabla donde cada fila representa un mes y las columnas son:

Columna	Tipo	Descripción
marketing_gasto	float	Gasto en marketing digital (MXN)
visitas_web	int	Número de visitas al sitio web
num_promociones	int	Cantidad de promociones activas
precio_promedio	float	Precio promedio de los productos
estacion	categoría	Verano, Otoño, Invierno, Primavera (one-hot)
ventas	float	Ventas totales del mes (variable objetivo)

Los datos se guardan, por ejemplo, en un archivo CSV (ventas_tienda.csv).

4.2. Pipeline de entrenamiento

1. Cargar los datos desde CSV.
2. Separar variables de entrada Xy variable objetivo y.
3. Dividir en conjuntos de **entrenamiento** y **prueba** (por ejemplo, 80 % / 20 %).
4. Codificar variables categóricas (one-hot encoding) si se requiere.
5. Entrenar un **RandomForestRegressor** sobre los datos de entrenamiento.
6. Evaluar con MAE, RMSE y R^2en el conjunto de prueba.

5. Resultados y evaluación

En la ejecución del modelo de **Bosques Aleatorios para regresión**, se obtienen las métricas MAE, RMSE y R^2.

Ejemplo de tabla de resultados (puedes rellenarla con los valores reales que te dé el código):

Métrica	*Valor*
MAE	XXXX.xx
RMSE	XXXX.xx

R^2	0.XX

Interpretación general:

- Un **MAE** bajo indica que, en promedio, el error absoluto de las predicciones es pequeño en comparación con la escala de las ventas.
- Un **RMSE** bajo (similar o ligeramente superior al MAE) indica que los errores grandes no son frecuentes.
- Un R^2cercano a 1 indica que una gran proporción de la variabilidad de las ventas es explicada por el modelo.

Si el desempeño no es satisfactorio (por ejemplo, R^2cercano a 0 o negativo, MAE muy alto), se podrían aplicar las siguientes mejoras:

- Recolectar más datos históricos.
- Incluir nuevas variables relevantes (competencia, campañas especiales, indicadores macroeconómicos).
- Ajustar hiperparámetros del Random Forest (n_estimators, max_depth, min_samples_split, etc.).
- Probar otros algoritmos (por ejemplo, Gradient Boosting, XGBoost).

6. Conclusiones y recomendaciones

En este trabajo se revisaron cuatro algoritmos de aprendizaje supervisado: dos para regresión (regresión lineal y bosques aleatorios) y dos para clasificación (regresión logística y bosques aleatorios para clasificación). Se analizaron sus objetivos, principios de funcionamiento, métricas de evaluación más utilizadas, así como sus principales fortalezas y limitaciones.

En el caso de estudio planteado, la **predicción de ventas mensuales** se abordó como un problema de regresión. Por las características del problema y la posible presencia de relaciones no lineales entre las variables explicativas y las ventas, se justificó el uso de un **Bosque Aleatorio de regresión**, que ofrece un buen compromiso entre desempeño y robustez.

La implementación en Python con scikit-learn permitió construir un pipeline completo que incluye preprocesamiento de datos, separación entrenamiento/prueba, entrenamiento del modelo y evaluación cuantitativa usando MAE, RMSE y R^2.

Como recomendaciones finales:

- Utilizar modelos simples (como regresión lineal) como línea base para comparar el desempeño de modelos más complejos.

- Realizar una validación más robusta (por ejemplo, validación cruzada) para estimar mejor la capacidad de generalización.
- Incluir análisis de importancia de variables para entender qué factores impactan más en las ventas y apoyar decisiones de negocio.

7. Referencias (formato APA)

(Ajusta el año/edición si usas otras versiones)

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R* (2nd ed.). Springer.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.