UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

Extracción de Conocimiento en Bases de Datos

III.1. Análisis Supervisado

IDGS91N

PRESENTA:

JORGE ALEJANDRO HERNANDEZ CONTRERAS

DOCENTE:

Enrique Mascote

Chihuahua, Chih.28 de Noviembre de 2025

Introducción

El objetivo de este trabajo es identificar y comprender el proceso de investigación e implementación de modelos supervisados (regresión y clasificación). Se muestra teoría, selección de algoritmos, diseño de un caso práctico, implementación en Python (scikit-learn), evaluación mediante métricas y conclusiones.

Investigación de algoritmos

Regresión

Regresión lineal

Qué resuelve:

Predice una variable continua como combinación lineal de las características.

Principio de funcionamiento:

Ajusta coeficientes para minimizar la suma de los residuos al cuadrado.

Métricas típicas:

MAE, MSE, RMSE, R^2.

Fortalezas:

Interpretabilidad, rápida, requiere pocos datos para converger.

Limitaciones:

Supone relación lineal; sensible a outliers; puede subajustar relaciones no lineales.

Random Forest Regressor

Qué resuelve:

Predicción de variables continuas usando ensambles de árboles.

Principio de funcionamiento:

Construye múltiples árboles de decisión en subconjuntos muestreados y promedia predicciones para reducir varianza.

Métricas típicas:

MAE, RMSE, R^2.

Fortalezas:

Maneja relaciones no lineales, resistente a outliers, poca preprocesamiento.

Limitaciones:

Menos interpretable; puede sobreajustar si no se regulan hiperparámetros; costoso en cómputo.

Clasificación

Regresión logística

Qué resuelve:

Clasificación binaria estimando la probabilidad de clases.

Principio de funcionamiento:

Modela la log-odds de la probabilidad de clase como función lineal de las features y aplica la función sigmoide.

Métricas típicas:

Accuracy, Precision, Recall, F1-score, AUC.

Fortalezas:

Rápida, probabilística, interpretable, robusta con regularización.

Limitaciones:

Supone separación lineal en las features; difícil manejar relaciones complejas sin ingeniería de características.

Random Forest Classifier

Qué resuelve:

Clasificación usando ensambles de árboles.

Principio de funcionamiento:

Votación de múltiples árboles entrenados en subconjuntos aleatorios; cada árbol aporta una predicción y la mayoría decide la clase.

Métricas típicas:

Accuracy, Precision, Recall, F1-score, matriz de confusión, AUC.

Fortalezas:

Maneja no linealidades, robusto a ruido, poca necesidad de escalado de features.

Limitaciones:
Menos interpretable, puede ser costoso, necesita ajuste de hiperparámetros.

Caso de estudio y justificación

Definición del problema

Regresión:
Predicción de ventas mensuales de una tienda (variable continua) a partir de variables históricas y promociones.
Clasificación:
Clasificar si un cliente realizará una compra (sí/no) el próximo mes.

Justificación del algoritmo elegido

Para la predicción de ventas (regresión) se elige Random Forest Regressor porque suele capturar relaciones no lineales entre variables sin necesidad de un modelado complejo de interacción; además maneja variables categóricas codificadas y es robusto a outliers. Para la clasificación de clientes se propone Logistic Regression como baseline por su interpretabilidad y rápida convergencia; como modelo secundario se probará Random Forest Classifier para mejorar performance si hay relaciones no lineales.

Diseño e implementación

Variables de entrada y estructura de datos

Regresión (ventas):

- Fecha
- Promoción
- Precio promedio
- Inventario disponible
- Publicidad
- Ventas históricas

Clasificación (cliente compra):

- Edad
- Género
- Última compra
- Número de visitas en el último mes
- Historial de compras
- Respuesta a promociones

Pipeline de entrenamiento

1. Limpieza: imputación de valores faltantes.
2. Feature engineering: variables temporales, lags, encoding categórico.
3. Escalado.
4. División: test y/o validación cruzada temporal para series.
5. Entrenamiento y ajuste de hiperparámetros.
6. Evaluación en test y análisis de residuos / matriz de confusión.

Implementación

```python
# Requisitos: scikit-learn, pandas, numpy
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn. (module) metrics t LinearRegression, LogisticRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix


def evaluar_regresion(y_true, y_pred):
mae = mean_absolute_error(y_true, y_pred)    Bloque con sangría previsto
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_true, y_pred)
return {'MAE': mae, 'MSE': mse, 'RMSE': rmse, 'R2': r2}
```

Resultados y evaluación

Explicar resultados esperados: comparar MAE/RMSE/R^2 entre modelos; para clasificación comparar accuracy/F1 y analizar la matriz de confusión para ver tipo de errores. Discutir si el modelo sufre overfitting.

Posibles mejoras

- Más datos o features.
- Ingeniería de features.
- Regularización.
- Ensambles y stacking.
- Ajuste de umbral para optimizar Precision/Recall según la aplicación.

Conclusión

El desarrollo de esta actividad permitió comprender de manera integral el proceso de selección, diseño e implementación de modelos supervisados para tareas de regresión y clasificación. La revisión teórica de distintos algoritmos evidenció que cada uno presenta fortalezas y limitaciones que deben evaluarse cuidadosamente según la naturaleza del problema y las características de los datos. Asimismo, el caso práctico mostró la importancia de construir un pipeline adecuado, que incluya limpieza, ingeniería de características, división de datos y validación, ya que estos pasos influyen directamente en el desempeño del modelo. La comparación de métricas permitió identificar cuáles algoritmos se ajustan mejor a los objetivos

planteados, reforzando la idea de que no existe una solución universal, sino modelos más apropiados según el tipo de relación entre variables y la complejidad del conjunto de datos. Finalmente, esta experiencia facilitó una comprensión más sólida del aprendizaje supervisado y reafirmó la relevancia de justificar cada decisión técnica dentro del proceso de modelado, con miras a desarrollar soluciones predictivas más confiables, interpretables y útiles en contextos reales.

Bibliografía

Pedregosa, F., et al. (s.f.). *LinearRegression — scikit-learn documentation*. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html?utm_source

scikit-learn developers. (s.f.). *RandomForestRegressor — scikit-learn documentation*. scikit-learn.https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?utm

Pedregosa, F., et al. (s.f.). *LogisticRegression — scikit-learn documentation*. scikit-learn.https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html?utm

scikit-learn developers. (s.f.). *RandomForestClassifier — scikit-learn documentation*. scikit-learn.https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?utm

scikit-learn developers. (s.f.). *Model evaluation — metrics and cross-validation (User Guide)*. scikit-learn.https://scikit-learn.org/stable/modules/model_evaluation.html?utm