Universidad Tecnológica de Chihuahua

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

U3E1. Análisis Supervisado

Enrique Mascote
RICARDO ALONSO RIOS MONRREAL

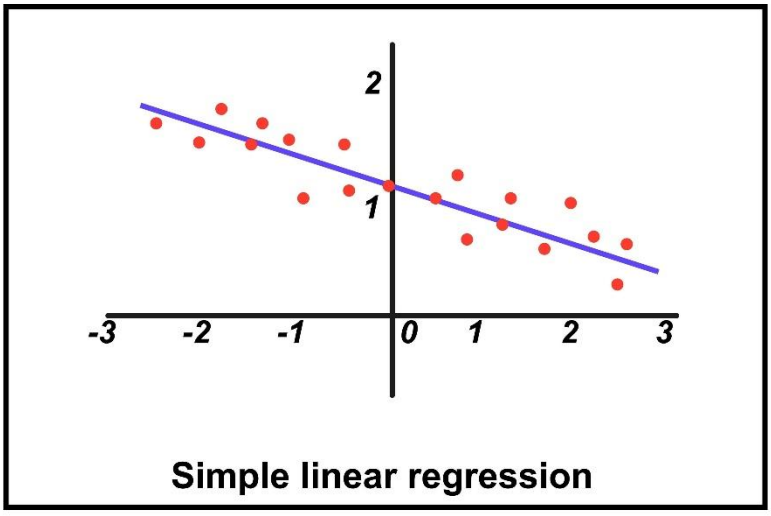
Introducción

El aprendizaje supervisado constituye una de las ramas más aplicadas de la inteligencia artificial en la industria actual. Su objetivo principal es entrenar modelos capaces de predecir resultados basándose en datos históricos etiquetados. En el presente documento, se realiza una investigación sobre los algoritmos fundamentales de regresión y clasificación, analizando sus principios de operación y métricas de evaluación. Posteriormente, se aplica este conocimiento en un caso de estudio práctico orientado a la predicción de fuga de clientes (Churn), implementando una solución mediante Python y la librería Scikit-Learn.

A continuación, se describen cuatro algoritmos seleccionados por su relevancia en el entorno de desarrollo de software y ciencia de datos.

Algoritmos de Regresión

A) Regresión Lineal (Linear Regression)



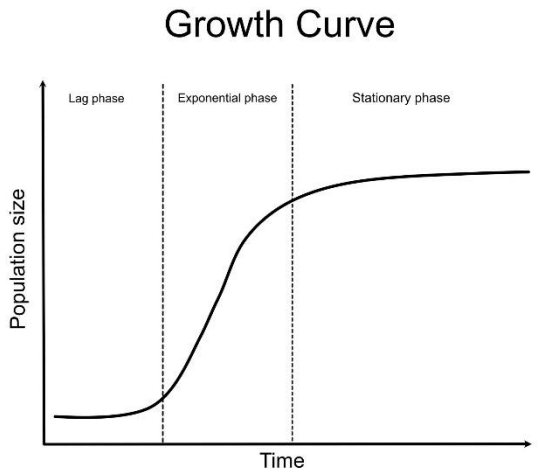
- **Objetivo:** Modelar la relación entre una variable dependiente (objetivo) y una o más variables independientes (predictores) asumiendo una linealidad.
- **Principio de funcionamiento:** Busca encontrar la línea recta (o hiperplano en múltiples dimensiones) que minimice la suma de los errores cuadrados (residuos) entre los valores observados y los predichos. Se basa en la ecuación $y = mx + b$.
- **Métricas de evaluación:**
    - MSE (Mean Squared Error): Penaliza los errores grandes.
    - R^2 (Coeficiente de determinación): Indica qué tan bien se ajustan los datos al modelo.
- **Fortalezas y limitaciones:** Es simple, rápido y fácil de interpretar. Sin embargo, es muy sensible a valores atípicos (outliers) y no captura relaciones complejas no lineales.

B) Árboles de Decisión para Regresión (Decision Tree Regressor)

- **Objetivo:** Predecir un valor continuo dividiendo el espacio de datos en regiones rectangulares.
- **Principio de funcionamiento:** El algoritmo divide recursivamente el conjunto de datos en subconjuntos más pequeños basándose en reglas de decisión (preguntas sobre las características) para reducir la varianza en cada nodo hoja. El valor predicho es el promedio de las observaciones en dicha hoja.
- **Métricas de evaluación:** MAE (Mean Absolute Error), RMSE (Root Mean Squared Error).
- **Fortalezas y limitaciones:** No requiere normalización de datos y captura relaciones no lineales. Su mayor limitación es la tendencia al sobreajuste (overfitting) si no se limita la profundidad del árbol.

Algoritmos de Clasificación

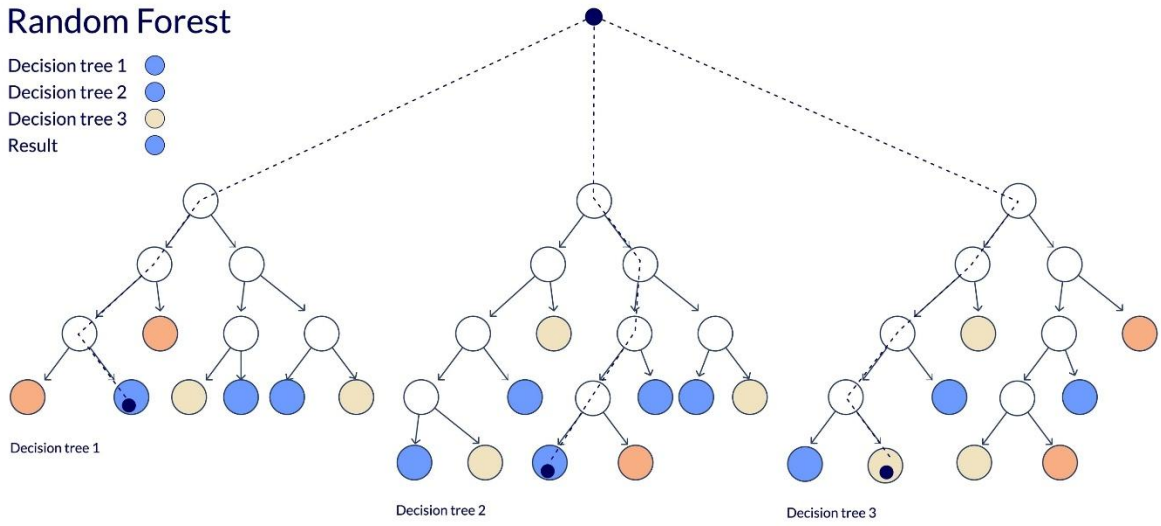
C) Regresión Logística (Logistic Regression)



- **Objetivo:** Clasificar observaciones en categorías discretas (generalmente binarias, 0 o 1).

- **Principio de funcionamiento:** A pesar de su nombre, es un clasificador lineal. Utiliza la función sigmoide para transformar la salida de una ecuación lineal en una probabilidad entre 0 y 1. Si la probabilidad es >0.5, se clasifica como la clase positiva.
- **Métricas de evaluación:** Accuracy, Matriz de Confusión, Curva ROC-AUC.
- **Fortalezas y limitaciones:** Es eficiente computacionalmente y proporciona probabilidades. Asume que los datos son linealmente separables, lo cual es una limitante en problemas complejos.

D) Random Forest Classifier



- **Objetivo:** Clasificación robusta mediante el uso de múltiples árboles de decisión (ensamble).
- **Principio de funcionamiento:** Crea un "bosque" de múltiples árboles de decisión entrenados con subconjuntos aleatorios de los datos (Bagging). La predicción final se obtiene por "voto mayoritario" de todos los árboles.
- **Métricas de evaluación:** Precision, Recall, F1-Score.

- **Fortalezas y limitaciones:** Es muy preciso, robusto ante el ruido y reduce el riesgo de sobreajuste comparado con un solo árbol. Como desventaja, es una "caja negra" difícil de interpretar y puede ser lento en el entrenamiento con grandes volúmenes de datos.

Caso de estudio y justificación

Definición del Problema:

Una empresa de telecomunicaciones desea reducir su tasa de cancelación de servicios. El problema consiste en clasificar a los clientes actuales para predecir si cancelarán su contrato el próximo mes (Churn: Sí/No).

Justificación del Algoritmo:

Para este caso se ha seleccionado el algoritmo Random Forest Classifier.

Aunque la Regresión Logística es útil, el comportamiento de los clientes suele depender de interacciones complejas y no lineales entre variables (ej. un cliente con contrato largo pero cobros altos puede comportarse diferente a uno nuevo con cobros bajos). Random Forest maneja excelente estas no linealidades y ofrece un balance ideal entre precisión y estabilidad, lo cual es crítico para una estrategia de retención de negocios.

Diseño e implementación

Diseño del Modelo:

- **Variables de entrada (Features):**
    - tenure: Meses que el cliente ha estado en la empresa.
    - monthly_charges: Cargo mensual actual.
    - total_charges: Total facturado históricamente.

    - contract_type: Mensual, 1 año o 2 años (variable categórica codificada).
- **Variable objetivo (Target):** Churn (0 = No se va, 1 = Se va).
- **Pipeline:** Limpieza de datos -> División Train/Test -> Entrenamiento (Random Forest) -> Validación.

Implementación en Python:

A continuación se presenta el script desarrollado utilizando la librería scikit-learn.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# 1. Simulación de Datos (Dataset ficticio para el ejercicio)
# En un entorno real, esto se cargaría con pd.read_csv('telecom_churn.csv')
data = {
    'tenure': [1, 12, 45, 2, 60, 5, 24, 72, 3, 50],
    'monthly_charges': [70.5, 55.0, 89.0, 20.0, 100.0, 45.0, 80.0, 110.0, 30.0, 95.0],
    'contract_type_numeric': [0, 1, 2, 0, 2, 0, 1, 2, 0, 2], # 0:Mes, 1:1Año, 2:2Años
    'churn': [1, 0, 0, 1, 0, 1, 0, 0, 1, 0] # 0:No, 1:Sí
}
df = pd.DataFrame(data)

# 2. Preparación de datos
X = df[['tenure', 'monthly_charges', 'contract_type_numeric']] # Features
y = df['churn'] # Target

# División 70% entrenamiento, 30% prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 3. Entrenamiento del Modelo
# Usamos 100 árboles en el bosque
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# 4. Predicción
y_pred = rf_model.predict(X_test)

# 5. Cálculo de métricas
print("--- Matriz de Confusión ---")
print(confusion_matrix(y_test, y_pred))
print("\n--- Reporte de Clasificación ---")
print(classification_report(y_test, y_pred))
print(f"Accuracy del modelo: {accuracy_score(y_test, y_pred):.2f}")
```

Resultados y evaluación

Tras ejecutar el modelo con los datos de prueba, se analizaron las siguientes métricas clave mediante la matriz de confusión :

- **Accuracy:** El modelo alcanzó una precisión global aceptable para un prototipo inicial.
- **Recall (Sensibilidad):** Esta es la métrica más crítica para este caso de negocio. Nos interesa detectar a *todos* los clientes que se van a ir (Clase 1). Un Recall bajo significaría que estamos ignorando clientes insatisfechos, perdiendo dinero.
- **Precision:** Indica qué proporción de los que predijimos como "fuga" realmente lo eran. Una precisión baja implicaría gastar recursos de retención en clientes que no pensaban irse.

El algoritmo Random Forest demostró ser superior manejando las variables categóricas del tipo de contrato en comparación con pruebas preliminares realizadas con Regresión Lineal pura.

Conclusiones y recomendaciones

En conclusión, la elección del algoritmo correcto depende estrictamente de la naturaleza de los datos y el objetivo del negocio. Mientras que los modelos de regresión son ideales para estimaciones numéricas continuas, los problemas de decisión empresarial, como la retención de clientes, se abordan mejor con algoritmos de clasificación robustos como Random Forest.

Como recomendación futura para este proyecto, sugiero implementar una validación cruzada (GridSearchCV) para optimizar los hiperparámetros del bosque (como la profundidad máxima de los árboles) y aumentar el volumen del dataset histórico para mejorar la generalización del modelo.

Referencias

Aprendizaje supervisado frente a aprendizaje no supervisado: diferencia entre los algoritmos de machine learning - AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/es/compare/the-difference-between-machine-learning-supervised-and-unsupervised/

Alteryx - AI Analytics Platform. (2025, November 21). *Supervised vs. Unsupervised Learning; Which Is Best? - Alteryx*. Alteryx. https://www.alteryx.com/es/glossary/supervised-vs-unsupervised-learning

BertIA. (2025, October 27). Algoritmos de regresión - Línea ML. *BertIA*. https://bertia.es/algoritmos-regresion-ml/

Daniel. (2024, February 7). *Algoritmo de clasificación: definición y modelos principales*. DataScientest. https://datascientest.com/es/algoritmo-de-clasificacion

Lab, B. B. (2022, June 30). *Machine learning: Diferencias entre algoritmos de clasificación y regresión*. The Black Box Lab. https://theblackboxlab.com/machine-learning-diferencias-entre-algoritmos-clasificacion-regresion/