Universidad Tecnológica de Chihuahua
Tecnologías de la Información

Análisis Supervisado

Alumno:

Jatzel Israel Cruz Castruita

Grupo:

IDGS91N

Materia:

Extracción de Conocimiento en Bases de Datos

Docente:

Enrique Mascote

Introducción

En la actualidad, el análisis de datos y el uso de algoritmos de aprendizaje automático se han convertido en herramientas fundamentales para resolver problemas reales en diversos sectores, como comercio, salud, finanzas y logística. Las organizaciones generan grandes volúmenes de información y requieren métodos eficientes para interpretarla, predecir comportamientos y tomar decisiones informadas. Dentro de este contexto, la regresión y la clasificación destacan como dos de las técnicas más utilizadas debido a su capacidad para modelar relaciones, identificar patrones y proporcionar soluciones basadas en datos.

El objetivo de este documento es presentar una investigación clara y estructurada sobre distintos algoritmos de regresión y clasificación, explicando su funcionamiento, métricas de evaluación y principales ventajas y limitaciones. Además, se desarrolla un caso de estudio práctico donde se implementa un modelo predictivo utilizando una metodología completa que abarca desde la selección del algoritmo hasta el análisis de sus resultados. Con ello, se busca comprender cómo estos métodos pueden aplicarse en situaciones reales y establecer las bases para futuras mejoras o aplicaciones más avanzadas dentro del campo del aprendizaje automático.

Algoritmo de regresión

Regresión Lineal

Qué resuelve

Predice valores numéricos continuos, como el precio de una casa, la temperatura o el nivel de ventas.

Principio de funcionamiento

Busca ajustar una línea recta o un plano cuando hay varias variables. El objetivo es minimizar la suma de los errores entre las predicciones y los valores reales mediante el método de mínimos cuadrados.

Métricas típicas de evaluación

Se evalúa con medidas como el error absoluto medio, el error cuadrático medio, la raíz del error cuadrático medio y el coeficiente de determinación R^2.

Fortalezas y limitaciones

Es un modelo fácil de entender y muy rápido de entrenar, especialmente útil cuando la relación entre variables es lineal.

Sin embargo, su desempeño disminuye cuando la relación entre los datos no es lineal. También es sensible a valores atípicos y a problemas de multicolinealidad entre variables.

Árboles de Decisión para Regresión

Qué resuelve

Predicen valores continuos dividiendo los datos en segmentos más pequeños según distintos umbrales.

Principio de funcionamiento

Construyen un árbol que realiza divisiones sucesivas basadas en valores de las variables. Cada división intenta reducir la varianza de los valores en los nodos. Al final, cada hoja del árbol representa un valor promedio que corresponde a la predicción.

Métricas típicas de evaluación

Se suelen medir mediante el error absoluto medio, el error cuadrático medio, la raíz del error cuadrático medio y el coeficiente de determinación.

Fortalezas y limitaciones

Son fáciles de interpretar, capaces de capturar relaciones no lineales y no requieren normalizar los datos.

Como desventajas, tienden a sobreajustarse si no se controlan sus profundidades, suelen ser inestables ante cambios en los datos y sus predicciones pueden ser menos precisas que las de modelos más avanzados.

Algoritmos de Clasificación

K-Nearest Neighbors (KNN)

Qué resuelve

Clasifica elementos asignándolos a la categoría mayoritaria entre los ejemplos más cercanos.

Principio de funcionamiento

Para clasificar un dato nuevo, se calcula su distancia con respecto a todos los datos del conjunto original. Se seleccionan los K vecinos más cercanos y la clase que más se repite entre ellos se convierte en la predicción final.

Métricas típicas de evaluación

Se evalúa mediante medidas como la exactitud, la puntuación F1, la precisión, el recall y la matriz de confusión.

Fortalezas y limitaciones

Es sencillo, intuitivo y adecuado cuando el conjunto de datos no es muy grande. Además, no requiere un entrenamiento tradicional.

Como limitaciones, se vuelve lento cuando hay muchos datos, es sensible a la escala de las variables y pierde rendimiento en espacios con muchas dimensiones.

Árboles de Decisión para Clasificación

Qué resuelve

Clasifica datos en categorías utilizando reglas simples basadas en condiciones sobre los atributos.

Principio de funcionamiento

Divide los datos mediante preguntas del tipo “el atributo X es mayor que un valor Y”. Cada división intenta aumentar la pureza del nodo utilizando criterios como la impureza de Gini o la entropía. Las hojas del árbol representan las clases finales.

Métricas típicas de evaluación

Normalmente se evalúan usando exactitud, precisión, recall, puntuación F1 y la matriz de confusión.

Fortalezas y limitaciones

Permiten una interpretación clara y visual, funcionan sin necesidad de un fuerte preprocesamiento y pueden capturar relaciones no lineales entre las variables.

Sus principales limitaciones son el sobreajuste, su sensibilidad a pequeñas variaciones en el conjunto de datos y que, en muchos casos, ofrecen menor precisión que modelos más complejos como Random Forest o XGBoost.

Sección 2: Solución de caso de estudio

Caso práctico

> Se busca predecir el nivel de ventas mensuales de una tienda de productos electrónicos. La empresa desea anticipar inventarios y planificar compras. Para ello, se cuenta con datos históricos que incluyen variables como el número de clientes atendidos, el gasto en publicidad, el mes del año y las ventas de meses anteriores.

Justificación del algoritmo elegido

- Para este caso se selecciona Regresión Lineal. Este algoritmo es adecuado porque:
- El objetivo es predecir un valor numérico continuo (ventas).
- Las relaciones entre algunas variables y las ventas suelen ser lineales, por ejemplo: más publicidad o más clientes suelen aumentar las ventas.
- Es un modelo interpretable, lo que permite a la empresa entender qué factores influyen más.
- Es rápido de entrenar y funciona bien como modelo base antes de probar opciones más complejas.

Diseño del modelo

Variables de entrada

- Clientes: número de clientes que entraron al negocio en el mes.
- Publicidad: cantidad invertida en marketing.
- Mes: número que representa el mes del año.
- Ventas del mes anterior: histórico inmediato.

Variable de salida

> Ventas del mes actual (valor numérico).

Estructura de datos

Los datos se almacenan en un DataFrame con columnas para cada una de las variables mencionadas. Cada fila representa un mes de operación.

Pipeline de entrenamiento

1. Cargar los datos.
2. Separar variables de entrada y salida.
3. Dividir en conjunto de entrenamiento y prueba.
4. Entrenar la Regresión Lineal.
5. Realizar predicciones.
6. Calcular métricas MAE, MSE y RMSE.
7. Analizar el rendimiento.

Análisis de resultados

El modelo entrega errores bajos, lo cual indica que la Regresión Lineal es capaz de capturar la relación entre la publicidad, los clientes, las ventas previas y las ventas actuales. El RMSE obtenido representa el promedio del error de predicción en unidades de ventas y es aceptable para este tipo de datos.

Posibles mejoras

- Incluir variables adicionales como promociones, días laborables del mes o estacionalidad más detallada.
- Probar modelos más complejos como Árboles de Decisión, Random Forest o XGBoost.
- Realizar normalización o estandarización si se usan algoritmos sensibles a la escala.
- Aplicar validación cruzada para mejorar la confiabilidad de las métricas.

Conclusión y recomendaciones

Después de analizar y aplicar los algoritmos de regresión y clasificación dentro del caso de estudio, puedo concluir que trabajar con modelos de aprendizaje automático no solo permite resolver problemas reales, sino también comprender mejor la dinámica de los datos y la forma en que cada variable influye en los resultados. Al implementar la Regresión Lineal en el escenario de predicción de ventas, confirmé que este modelo es una excelente base para iniciar, ya que ofrece interpretaciones claras y un rendimiento adecuado cuando las relaciones entre las variables son relativamente sencillas. Esto me permitió obtener una visión más precisa del comportamiento de las ventas y entender qué aspectos aportan mayor peso a la predicción.

Sin embargo, reconozco que ningún modelo es perfecto y que siempre existen oportunidades de mejora. Por ello, considero importante seguir explorando alternativas más complejas como Árboles de Decisión, Random Forest o incluso modelos basados en técnicas de ensamble que podrían aumentar la precisión y la robustez del sistema. También creo que la calidad de los datos es un factor determinante, por lo que recomendaría profundizar en la recolección, limpieza y generación de nuevas variables que capturen mejor el contexto real del problema.

Referencias

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (2.ª ed.). O'Reilly Media.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2.ª ed.). Springer.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Anexos

C: > Users > Jatze > OneDrive > Documentos > prueba.py > ...

```python
# Importacion de librerias
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Ejemplo de datos simulados
data = {
    "Clientes": [200, 250, 300, 280, 320, 350, 400, 380, 420, 450],
    "Publicidad": [5000, 5500, 6000, 6200, 6500, 7000, 7200, 7400, 7800, 8000],
    "Mes": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    "Ventas_previas": [30000, 32000, 34000, 33000, 36000, 38000, 40000, 39000, 42000, 44000],
    "Ventas": [32000, 34000, 36000, 35000, 38000, 40000, 42000, 41000, 44000, 46000]
}

df = pd.DataFrame(data)

# Variables independientes y dependiente
X = df[["Clientes", "Publicidad", "Mes", "Ventas_previas"]]
y = df["Ventas"]

# Division de datos (80% entrenamiento, 20% prueba)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Entrenamiento del modelo
modelo = LinearRegression()
modelo.fit(X_train, y_train)

# Predicciones
predicciones = modelo.predict(X_test)

# Calculo de metricas
mae = mean_absolute_error(y_test, predicciones)
mse = mean_squared_error(y_test, predicciones)
rmse = np.sqrt(mse)

print("Error Absoluto Medio (MAE):", mae)
print("Error Cuadratico Medio (MSE):", mse)
print("Raiz del Error Cuadratico Medio (RMSE):", rmse)
```