

Práctica 3

Regresión y categorización



Profesor:
David Campoy Miñarro

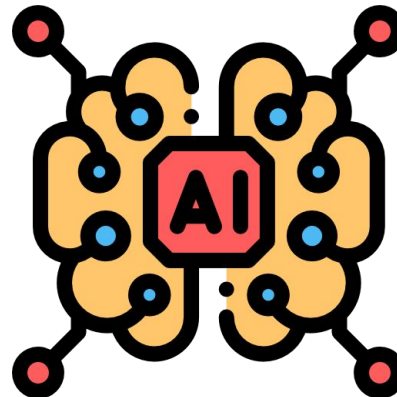


Regresión y categorización

Objetivo: Conocer las diferencias entre regresión y categorización.

En el campo del *machine learning* los algoritmos se pueden dividir en: regresión y clasificación.

La **regresión** es un tipo de problema en el aprendizaje automático en el que se busca predecir un valor numérico o continuo basado en datos previos. Se trata de encontrar una relación matemática entre variables de entrada y salida.



La **clasificación** es otro tipo de problema en el aprendizaje automático donde se asignan etiquetas o categorías a un conjunto de datos. El objetivo es predecir a qué clase o categoría pertenecen los datos, es decir, separar los datos en diferentes grupos o clases.



MACHINE LEARNING: DIFERENCIAS ENTRE ALGORITMOS DE CLASIFICACIÓN Y REGRESIÓN

<https://theblackboxlab.com/2022/05/06/machine-learning-diferencias-entre-algoritmos-clasificacion-regresion/>



Prueba el siguiente código

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC

# Generar datos aleatorios para regresión
np.random.seed(0)
X_regression = np.random.uniform(0, 10, 100)
y_regression = 2 * X_regression + 3 + np.random.normal(0, 2, 100)

# Generar datos aleatorios para clasificación
np.random.seed(1)
X_classification_1 = np.random.normal(3, 1, 100)
y_classification_1 = np.random.normal(3, 1, 100)

X_classification_2 = np.random.normal(7, 1, 100)
y_classification_2 = np.random.normal(7, 1, 100)

# Crear modelos
regression_model = LinearRegression()
classification_model = SVC(kernel='linear')

# Entrenar modelos
X_regression = X_regression.reshape(-1, 1)
regression_model.fit(X_regression, y_regression)
```

¿Qué hace el programa?

Se generan datos aleatorios para representar dos escenarios diferentes: uno para regresión y otro para clasificación.

Para la regresión, se crean puntos (X, Y) donde Y sigue una relación lineal con X ($Y = 2 \cdot X + 3$) con algo de ruido aleatorio añadido para simular datos del mundo real.

Para la clasificación, se generan dos conjuntos de puntos en un plano 2D, cada uno centrado alrededor de (3, 3) y (7, 7) respectivamente.

Modelos IA utilizados:

- `LinearRegression()` es un modelo de regresión lineal que intenta encontrar la mejor línea.
- `SVC(kernel='linear')` es un modelo de clasificación SVM.

Prueba el siguiente código

```
X_classification = np.concatenate((np.stack((X_classification_1, y_classification_1), axis=1),
                                     np.stack((X_classification_2, y_classification_2), axis=1)))
y_classification = np.concatenate([np.zeros(100), np.ones(100)])
classification_model.fit(X_classification, y_classification)

# Crear puntos para graficar las predicciones
x_values = np.linspace(0, 10, 100).reshape(-1, 1)
y_values_regression = regression_model.predict(x_values)

xx, yy = np.meshgrid(np.linspace(0, 10, 100), np.linspace(0, 10, 100))
xy = np.column_stack([xx.ravel(), yy.ravel()])
y_values_classification = classification_model.predict(xy).reshape(xx.shape)

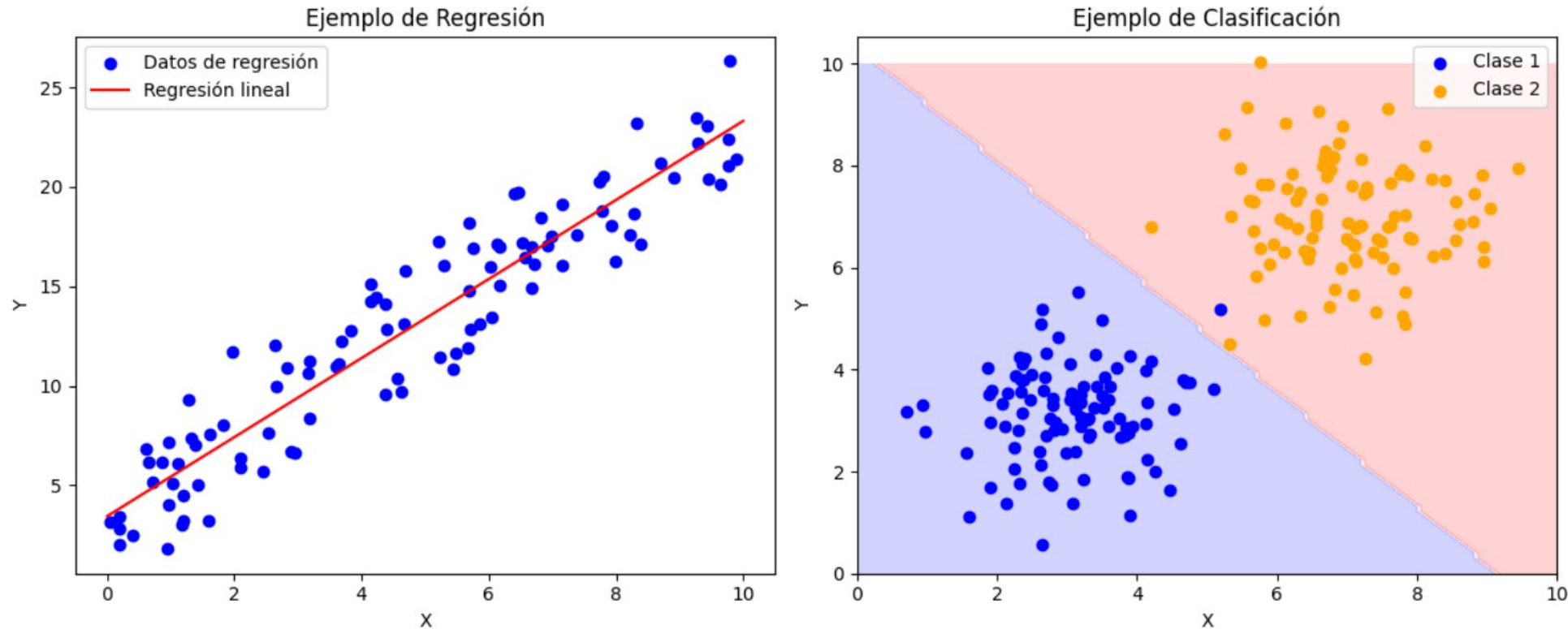
# Graficar resultados de regresión y clasificación
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.scatter(X_regression, y_regression, label='Datos de regresión', color='blue')
plt.plot(x_values, y_values_regression, color='red', label='Regresión lineal')
plt.title('Ejemplo de Regresión')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

```
plt.subplot(1, 2, 2)
plt.contourf(xx, yy, y_values_classification,
             alpha=0.2, cmap='bwr')
plt.scatter(X_classification_1,
            y_classification_1, label='Clase 1',
            color='blue')
plt.scatter(X_classification_2,
            y_classification_2, label='Clase 2',
            color='orange')
plt.title('Ejemplo de Clasificación')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()

plt.tight_layout()
plt.show()
```

Resultado del aprendizaje

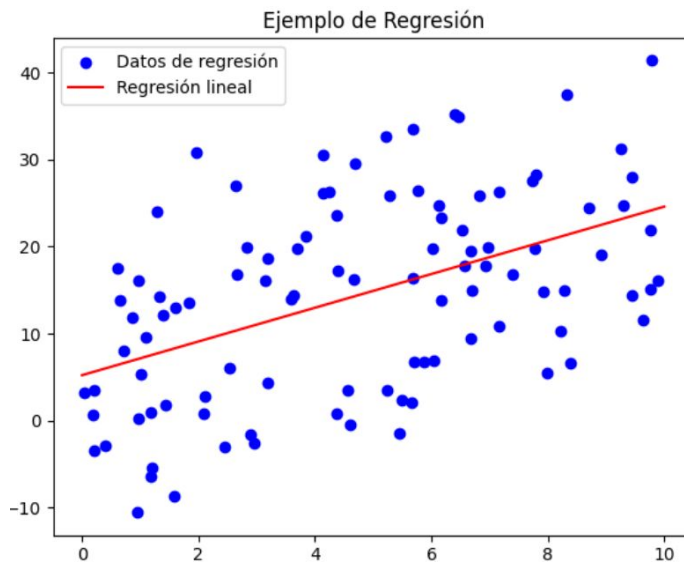


- El modelo de regresión ha sido capaz de encontrar la tendencia fijada por $Y = 2 \cdot X + 3$
- El modelo de clasificación ha sido capaz de encontrar la línea que separa las fuentes de puntos.

Actividades

Modifica las relaciones lineales para la regresión:

- $Y = 3X + 3$
- $Y = -2 \cdot X + 3$
- $Y = 2 \cdot X - 2$
- $2 * \text{np.sin}(0.5 \cdot X_{\text{regression}} + (\text{np.pi}/4)) + 3$
- Modifica el rango aleatorio
- Prueba otras funciones



Para la clasificación:

- Añade una tercera fuente de puntos.
- Añade una cuarta fuente de puntos.
- Varía el componente aleatorio.

Ayuda:

```
y_classification = np.concatenate([np.zeros(100), np.ones(100), np.full(100, 2)])
```

