

The background is a dark navy blue. It features several large, overlapping, semi-transparent geometric shapes in various colors: bright green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to be layered on top of others. The overall aesthetic is modern and abstract.

Regresión Lineal

Adrian Ureña M. B57340

Regresión Lineal

La regresión lineal es un método estadístico que se utiliza para modelar la relación entre una variable dependiente (la que queremos predecir) y una o más variables independientes (las que usamos para hacer la predicción). Se llama "lineal" porque asume que la relación entre las variables puede aproximarse mediante una línea recta.

```
1 import numpy as np
2 from sklearn.model_selection import train_test_split
3 from sklearn.metrics import mean_squared_error, r2_score
4 from sklearn.linear_model import LinearRegression
5 import matplotlib.pyplot as plt
```

- ❑ **numpy:** Es una biblioteca de Python que proporciona soporte para trabajar con arreglos multidimensionales y matrices.
- ❑ **sklearn.model_selection:** Es un módulo de scikit-learn (también conocido como sklearn) que proporciona herramientas para dividir conjuntos de datos en subconjuntos de entrenamiento y prueba de forma aleatoria.
- ❑ **sklearn.metrics:** Es un módulo de scikit-learn que contiene funciones para evaluar la precisión y el rendimiento de los modelos de aprendizaje automático.
- ❑ **mean_squared_error:** Es una función dentro de sklearn.metrics que calcula el error cuadrático medio entre las predicciones de un modelo y los valores reales.
- ❑ **matplotlib.pyplot:** Es un módulo de la biblioteca matplotlib que proporciona una interfaz para crear gráficos y visualizaciones en Python.

```
25  np.random.seed(0)
26  x = 2* np.random.randn(100,1)
27  y = 5 + 3 * x + np.random.randn(100,1)
```

`np.random.seed(0)`: Establece la semilla (seed) para el generador de números aleatorios de la biblioteca NumPy en 0.

X es la variable es la variable independiente

Y es la variable dependiente

`np.random.randn(100, 1)`: Genera una matriz de 100 filas y 1 columna de números aleatorios distribuidos normalmente con media 0 y desviación estándar 1.

```
30 # Dividir los datos en conjuntos de entrenamiento y prueba
31 X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=42)
```

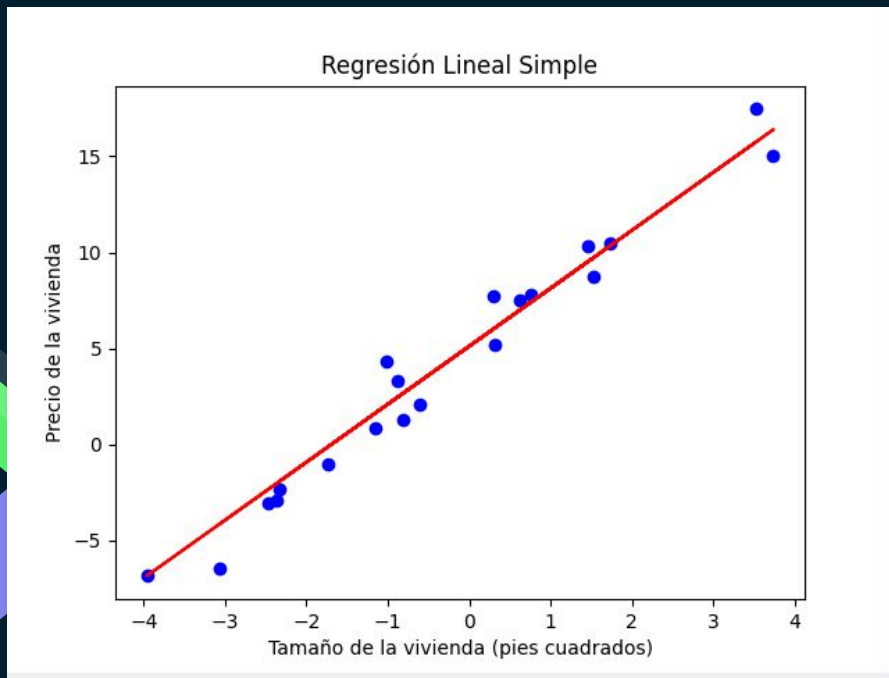
La línea de código `X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)` divide los conjuntos de datos `x` e `y` en un conjunto de entrenamiento y un conjunto de prueba. El 80% de los datos se utilizan para entrenar el modelo (`X_train, y_train`), mientras que el 20% se reserva para evaluar el modelo (`X_test, y_test`). La división se realiza aleatoriamente, pero el parámetro `random_state=42` asegura que la división sea reproducible, lo que significa que obtendremos la misma división cada vez que ejecutemos el código con la misma semilla.

```
33 # Crear el modelo de regresión lineal
34 modelo = LinearRegression()
35 # Entrenar el modelo con los datos de entrenamiento
36 modelo.fit(X_train, y_train)
```

Esta línea de código crea un modelo de regresión lineal utilizando la clase `LinearRegression()` de la biblioteca `sklearn.linear_model` y luego lo entrena con los datos de entrenamiento proporcionados (`X_train` y `y_train`)

```
41 y_pred = modelo.predict(x)
42
43 print("MSE:", mean_squared_error(y,y_pred))
44 print("R2:", r2_score(y,y_pred))
```

Primero, el método `predict()` del modelo se utiliza para predecir los valores de la variable dependiente (`y_pred`) basados en las variables independientes (`x`). Luego, se calculan dos métricas de evaluación: el error cuadrático medio (MSE) y el coeficiente de determinación (R cuadrado). El MSE mide cuán cerca están las predicciones del modelo de los valores reales, mientras que el R cuadrado indica qué tan bien el modelo se ajusta a los datos.



Interpretación:

En este caso, la línea de regresión está sumamente cerca de los puntos dados por lo que se acerca a los valores reales del modelo. El R^2 está en 0.97 lo que está muy cerca de 1 lo que indica que explica gran parte de los datos del modelo.