



MODELO DE REGRESIÓN LINEAL

FIGURELLA MORA NAVARRO



```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```


- NumPy lo utilizamos para generar números random o hacer operaciones matemáticas
- LinearRegression crea el modelo de regresion lineal
- Sklearn.metrics importa funciones para poder evaluar el modelo de regresion lineal
- Matplotlib se utiliza para graficar el modelo

OPCIÓN 1 (DATOS RAMDON)

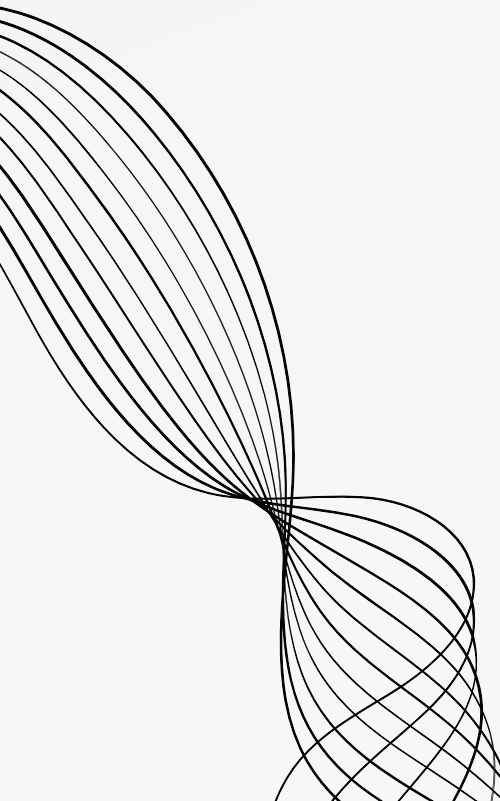
```
# Generar datos aleatorios para la variable independiente
np.random.seed(0)
X = np.random.rand(100, 1) # Variable independiente entre 0 y 1
y = 2 + 3 * X + np.random.randn(100, 1)
```

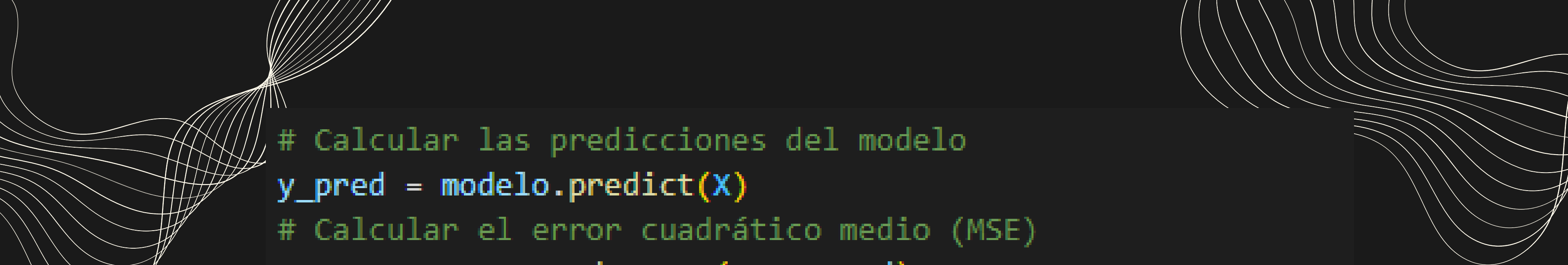
OPCIÓN 2 (DATOS ESPECÍFICOS)

```
# Datos de ejemplo: tamaño de la vivienda en pies cuadrados y precios en dólares
tamanos = np.array([600, 800, 1000, 1200, 1500]).reshape(-1, 1)
precios = np.array([100000, 150000, 200000, 230000, 300000])
```



```
# Crear el modelo de regresión lineal
modelo = LinearRegression()
# Entrenar el modelo con los datos
modelo.fit(X, y)
```

- En este paso, se entrena el modelo con los valores de las variables respectivamente, para que esté listo para hacer predicciones y ser graficado
- 



```
# Calcular las predicciones del modelo
y_pred = modelo.predict(X)
# Calcular el error cuadrático medio (MSE)
mse = mean_squared_error(y, y_pred)
# Calcular el coeficiente de determinación (R-cuadrado)
r2 = r2_score(y, y_pred)
```

- Calcula el error cuadrático medio (MSE) entre los valores reales de y y las predicciones
- Calcula el coeficiente de determinación (R-cuadrado) entre los valores reales de y y las predicciones y_pred

Impresión de resultados y del modelo de regresión

```
# Imprimir los resultados
print(f"Error Cuadrático Medio (MSE): {mse:.2f}")
print(f"Coeficiente de determinación (R^2): {r2:.2f}")
print(f"La predicción para X=0.5 es: {y_pred[0][0]:.2f}")

# Graficar los datos y la línea de regresión
plt.scatter(X, y, color='blue', alpha=0.5)
plt.plot(X, modelo.predict(X), color='red', linewidth=2)
plt.xlabel('Tamaño de la casa')
plt.ylabel('Precio de la casa')
plt.title('Regresión Lineal Simple')
plt.show()
```




¡GRACIAS!