Etapa 2 Diseño de sistemas de aprendizaje automatico

Nombre del estudiante:  
Jhon Edison Betancurt Lora

Grupo:

**Análisis de datos\_202016908A\_1394**

Tutor

Sandra Milena Patiño

Universidad Nacional Abierta y a Distancia-UNAD

Escuelas de Ciencias Básicas Tecnologías e Ingeniería

Programa de Ingeniería de sistemas

Quimbaya – Septiembre del 2023

**Regresión lineal**

**Codigo**

import pandas as pd

import numpy as np

import os

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

reglineal = pd.read\_csv("data.csv")

reglineal.head(5)  
  
x = reglineal[['metro']]

y = reglineal[['precio']]

model = LinearRegression()

model.fit(x, y)  
  
predi\_reg\_lineal = model.predict(x)

plt.scatter(x, y, color='blue')

plt.plot(x, predi\_reg\_lineal, color='red')

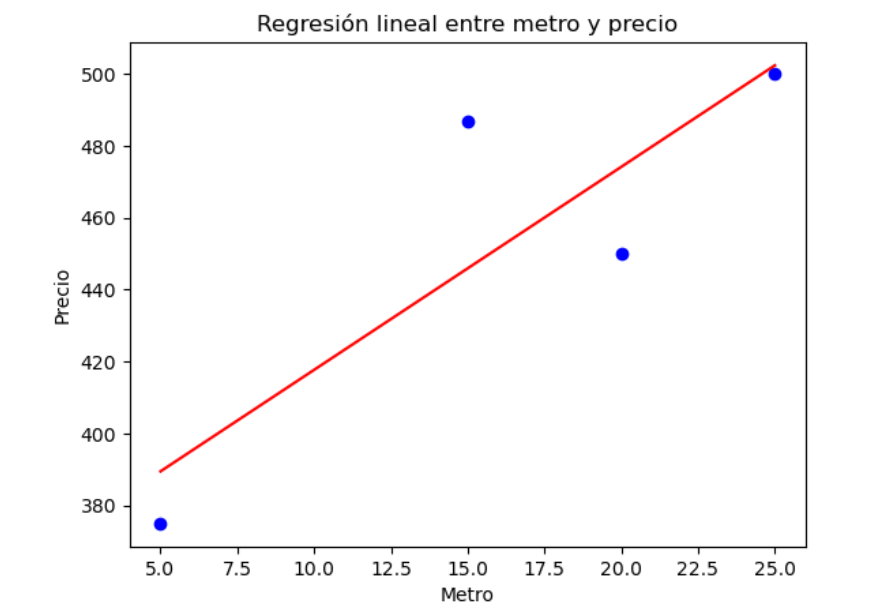
plt.xlabel('Metro')

plt.ylabel('Precio')

plt.title('Regresión lineal entre metro y precio')

plt.show()

**Grafico**

****

**Análisis**

Hay una relación positiva entre el valor de metros y el de precio, (esto se sabe por que la pendiente es positiva) entonces esto significa que se espera que el precio aumente cuando aumente el valor de metros.

Para el R-cuadrado me lo da el mismo código utilizando

print("R-cuadrado: ", model.score(x, y))

lo cual me da un valor aproximado a 0.737 lo que me dice que es bueno el modelo para hacer prediciones con los datos que se le dio.

Ya para el R-cuadrado ajustado se le aplica la formula

donde n es el numero de muestras en este caso es 4 y p es el numero de los predictores en este caso es 1, reemplazando R-cuadrado tenemos que el valor de = 0.6055