ESTADISTICA AVANZADA

PROF. JUAN IVAN NIETO HIPOLITO

REPORTE DE PRACTICA
REGRESIÓN LINEAL SIMPLE, R CUADRADA Y NORMALIDAD DE LOS RESIDUOS

KEVIN ALEJANDRO GONZALEZ TORRES
GRUPO 932

CÓDIGO

1.- DECLARAR LIBRERIAS E INGRESAR DATOS

- numpy (alias np) para operaciones numéricas.
- matplotlib.pyplot (alias plt) para crear gráficos y visualizaciones.
- r2_score de sklearn.metrics para calcular el puntaje r-cuadrado.
- pandas (alias pd) para manipulación de datos (aunque no se utiliza en el código)

Almacenamos los datos de las variables dependientes (y) e independientes (x), respectivamente.

```
10  import numpy as np
11  import matplotlib.pyplot as plt
12  from sklearn.metrics import r2_score
13  import pandas as pd
14
15  # Data
16  y_values = [790, 1160, 929, 865, 1140, 929, 1109, 1365, 1112, 1150, 980, 990, 1112,
17              1252, 1326, 1330, 1365, 1280, 1119, 1328, 1584, 1428, 1365, 1415, 1415,
18              1465, 1490, 1725, 1523, 1705, 1605, 1746, 1235, 1390, 1405, 1395]
19
20  x_values = [99, 95, 95, 90, 105, 105, 90, 92, 98, 99, 99, 101, 99, 94, 97, 97, 99, 104, 104,
21              105, 94, 99, 99, 99, 99, 102, 104, 114, 109, 114, 115, 117, 104, 108, 109, 120]
```

2.- ENCONTRAR R^2 E IMPRIMIR EL HISTOGRAMA

Calculamos un modelo de regresión polinómica de grado 9 (9 representa el grado del polinomio). Ajusta el modelo a los puntos de datos (x_values, y_values) utilizando np.polyfit y luego crea una función de predicción usando np.poly1d.

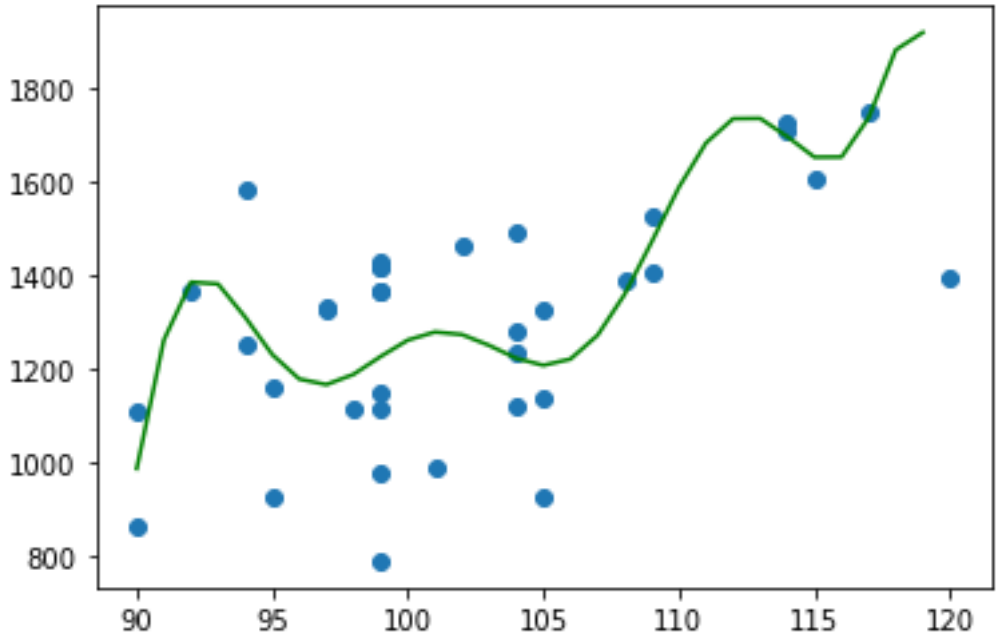
Utilizamos la función de predicción para generar valores y predichos (y_predicted) basados en los valores x en el conjunto de datos.

Por ultimo calculamos r-cuadrado, que mide la bondad de ajuste del modelo de regresión a los datos. Imprime r-cuadrado en la consola.

```
27  # Predicted values
28  y_predicted = prediction_function(x_values)
29
30  # R-squared score
31  r_squared = r2_score(y_values, y_predicted)
32  print("R-squared score: ", r_squared)
33
34  # Generate points for the regression line
35  x_regression = range(90, 120)
36  y_regression = prediction_function(x_regression)
37
38  # Scatter plot of the data
39  plt.scatter(x_values, y_values)
40  # Plot the regression line
41  plt.plot(x_regression, y_regression, c="g")
```

3.- IMAGEN DE LA GRAFICA OBTENIDA

Creamos el gráfico de dispersión de los puntos de datos originales, donde x_values se representa en el eje x y y_values en el eje y.



4.- OBTENER MEDIA Y DESVIACION ESTANDAR DEL ERROR

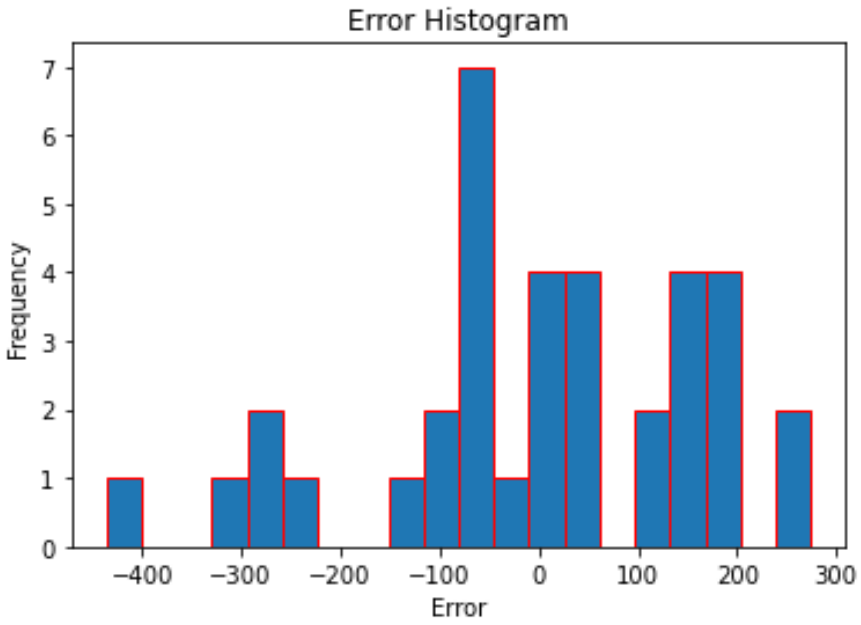
Calculamos los errores entre los valores y reales y los valores y predichos. Imprime los errores en la consola.

```
43  # Calculate errors
44  errors = [actual - predicted for actual, predicted in zip(y_values, y_predicted)]
45  print("Errors: ", errors)
46
47  # Mean error
48  mean_error = np.mean(errors)
49  print("Mean error: ", mean_error)
50
51  # Standard deviation of errors
52  std_error = np.std(errors)
53  print("Standard deviation of errors: ", std_error)
```

5.- IMPRIMIR HISTOGRAMA DE LOS ERRORES

Esta sección crea un histograma de los errores. Representa la distribución de los errores utilizando 20 contenedores (bins) y muestra el histograma.

```
55    # Error histogram
56    plt.figure()
57    plt.title("Error Histogram")
58    plt.xlabel("Error")
59    plt.ylabel("Frequency")
60    plt.hist(errors, bins=20, edgecolor='red')
61    plt.show()
62
```



6.- RESULTADOS

Finalmente, mostramos todos los resultados destinados a imprimirse.

```
In [3]: runfile('G:/My Drive/UABC/TercerSemestre/KAGT_Estadistica_Avanzada_932/
Practica_6_y_7/Practica_6_y_7.py', wdir='G:/My Drive/UABC/TercerSemestre/
KAGT_Estadistica_Avanzada_932/Practica_6_y_7')
R-squared score:  0.5144012943820622
Errors:  [-435.9610595703125, -70.0040283203125, -301.0040283203125,
-121.67333984375, -67.0787353515625, -278.0787353515625, 122.32666015625,
-19.4383544921875, -75.8236083984375, -75.9610595703125, -245.9610595703125,
-288.0977783203125, -113.9610595703125, -57.9404296875, 160.0758056640625,
164.0758056640625, 139.0389404296875, 57.95751953125, -103.04248046875,
120.9212646484375, 274.0595703125, 202.0389404296875, 139.0389404296875,
189.0389404296875, 189.0389404296875, 192.3446044921875, 267.95751953125,
30.23876953125, 52.173828125, 10.23876953125, -45.9542236328125, 11.086181640625,
12.95751953125, 31.4256591796875, -65.826171875, -0.8367919921875]
Mean error:  -0.016910129123263888
Standard deviation of errors:  166.3639300643938
```

CONCLUSIÓN

Como resultado, obtuvimos que r cuadrada es igual a 0.514401089.

La media es igual a -0.005645 y su desviación estándar 166.3639.

Dado que ya hemos calculado la media y la desviación estándar de los errores en el código proporcionado, podemos usar estos valores para obtener una indicación de si los errores se asemejan a una distribución normal. Sin embargo, la mejor manera de verificar esto sería visualmente, observando el histograma.

Si el histograma de errores muestra una forma de campana, simetría y una distribución centrada alrededor de la media de los errores, es más probable que los errores se aproximen a una distribución normal.

Cabe aclarar que todas estas funcionalidades son escenciales para empezar con el machine learning, cosa la cual me interesó bastante ya que básicamente es a lo que me quiero dedicar en mi formación profesional.