
CIENCIA DE DATOS EN PYTHON
HOJA DE TRABAJO # 3 - DICCIONARIOS, SETS Y PROGRAMACIÓN ORIENTADA A OBJETOS

Entregable: Su entregable deberá ser un link al repositorio de git que contiene el Jupyter Notebook donde resolvió los ejercicios planteados.

Parte #1 - (Programación Orientada a Objetos): Genere una clase llamada `my_regresion` en Python puro (sin utilizar ninguna librería) que sea capaz de encontrar de forma analítica los coeficientes de una regresión lineal (β_0 y β_1). Su clase debe poseer los siguientes campos: β_0 , β_1 los cuales representan los parámetros de la regresión, X y Y los cuales representan la lista de valores con los que se calcularon los parámetros de la regresión. Adicionalmente su clase debe poseer los siguientes métodos:

1. **Constructor Principal:** Este constructor debe poseer la siguiente forma `__init__(self, x, y)` donde X y Y son listas de números con la misma cantidad de elementos. A partir de este constructor usted deberá calcular los parámetros β_0 y β_1 puede usar las siguientes ecuaciones para este calculo:

$$\beta_0 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x^2 - (\sum x)^2}$$

$$\beta_1 = \frac{n \sum x_i^2 \sum y_i - \sum x_i \sum y_i x_i}{n \sum x^2 - (\sum x)^2}$$

2. **Método Predict(val):** Este método devuelve la predicción de la regresión aplicada sobre el valor que recibe como parámetro.
3. **Método PredictN(vals):** Este método devuelve la predicción de la regresión aplicada sobre una lista de valores que recibe como parámetro.
4. **Método r2():** Este método calcula el r^2 (coeficiente de determinación) de la regresión, para calcular dicho valor puede usar la siguiente ecuación:

$$r^2 = \frac{\sum_{i=1}^n (\hat{Y} - \bar{Y})^2}{\sum_{i=1}^n (\hat{Y} - Y)^2}$$

5. **Método r():** Este método devuelve el coeficiente de correlación el cual se calcula como la raíz cuadrada el coeficiente de determinación es decir:

$$r = \sqrt{r^2}$$

6. **Método getAllRegParams():** Este método devuelve un diccionario con todos los parámetros y estadísticos de la regresión lineal, el diccionario a devolver debe tener la siguiente estructura:

$$\{b0 : valor, b1 : valor, r2 : valor, r : valor\}$$

7. **Constructor from_file(fileName):** Este método es otro constructor el cual permite crear la regresión lineal a partir de un archivo .csv, notar que fileName es el nombre del archivo que recibe como parámetro. Dentro del archivo deben existir dos columnas, la primera es la variable dependiente es decir Y y la segunda se refiere a los valores de la variable independiente, es decir X regresión desde dos listas numéricas. Para desarrollar este método **NO** puede utilizar ninguna librería que lea un archivo .csv o similares.

A continuación se le muestra un esqueleto de la clase:

```
import numpy as np

class regresion():
    def __init__(self, x, y):
        self.beta0 = # calcular beta0 utilizando x & y
        self.beta1 = # calcular beta1 utilizando x & y
        self.x = #lista de valores de x
        self.y = #lista de valores de y

    def predict(self, val):
        #codigo

    def predictN(self, vals):
        #codigo

    def r2(self):
        #codigo

    def r(self):
        #codigo

    getAllParams(self):
        #codigo

    @classmethod
    def from_file(cls, path):
        #codigo
```