

Resumen IA FUNDAMENTALS

Regresion Lineal:

- La Regresion Cuantifica la Naturaleza de la Relacion
- La Correlacion mida la Fuerza de Asociacion entre las 2 Variables

Conceptos Basicos:

- Response (Dependent Variable, "Y" Variable, Target o Salida) : Variable a predecir
- Independent Variable (Feature, Atributo, "X" Variable) : La Variable que usamos para predecir la Respuesta
- Record (Row, case, instance, example): El vector de variable predictorias y los Valores de salida de un individuo específico o caso
- Intercept(β_0 , Ordenada a la Origen) : La intercepcion de la linea de Regresion, es decir, el valor pronosticado
- Regresion Coefficient (slope , β_1 ,weights , peso) : la Pendiente de la linea de Regresion
- Fitted Values: los estimados obtenidos de la linea de Regresion
- Residuals (Errors) : La diferencia entre los valores observados y los valores Ajustados (Fitted Values)
- Least squares(Minimos Cuadrados) : El Metodo de ajustar una regresion minimizando la suma de residuos

Ecuacion General de la Regresion Lineal Simple

$$Y = \beta_0 + \beta_1 X$$

- β_0 --> Intercept o interseccion
- β_1 --> Slope o Pendiente (Coeficiente de Regresion)
- Y --> Variable Dependiente
- X --> Vector de Caracteristicas (Variable Predictora)

Primer modelo de Regresion Lineal Simple

```
In [1]: #importamos las Libreias
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
import numpy as np
```

```
In [2]: #Cargamos los datos en la variable "data"
data=pd.read_csv('datasets\LungDisease.csv')
data
```

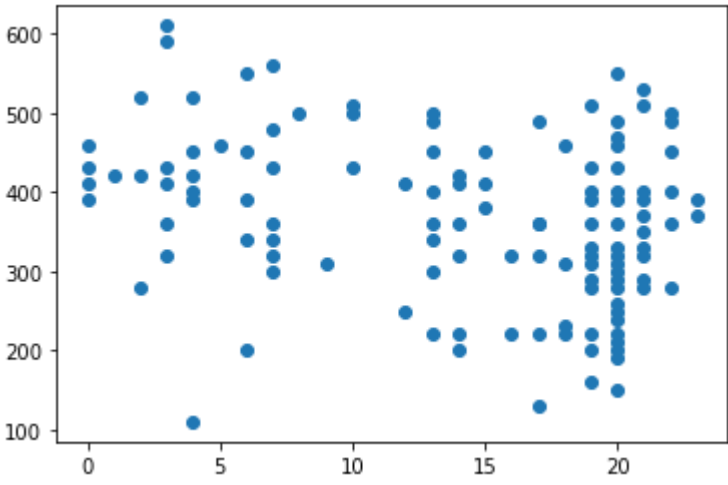
Out[2]:

	PEFR	Exposure
0	390	0
1	410	0
2	430	0
3	460	0
4	420	1
...
117	450	22
118	490	22
119	500	22
120	370	23
121	390	23

122 rows × 2 columns

```
In [3]: #Creamos un Scatter Plot pasando los valores "exposure" y "PEFR" dentro de data
plt.scatter(data.Exposure, data.PEFR)
```

Out[3]: <matplotlib.collections.PathCollection at 0x1bc427a8f70>



```
In [4]: #Creamos el Modelo de regresion Lineal, lo asignamos a la variable "model"
model = linear_model.LinearRegression()
model.fit(data[['Exposure']], data['PEFR']) # lo entrenamos con .fit y le pasamos los datos
```

Out[4]: <linear_regression.LinearRegression()

```
In [5]: #imprimimos los valores de Intercept y Coefficient (β0 y β1)
print("intercept: ", model.intercept_)
print("Coefficient: ", model.coef_[0])

intercept: 424.582806573957
coefficient: -4.1845764854614425
```

```
In [6]: #Creamos un Array desde el 0 a el 25
x = [ i for i in range(26)]
#lo trasformamos en un array de numpy para podes usar Ufuncs
x = np.array(x)
x
```

Out[6]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25])

```
In [7]: #Creamos las Variables Intecept y Coefficient con los valores obtenidos del modelo
intercept = model.intercept_
coefficient = model.coef_

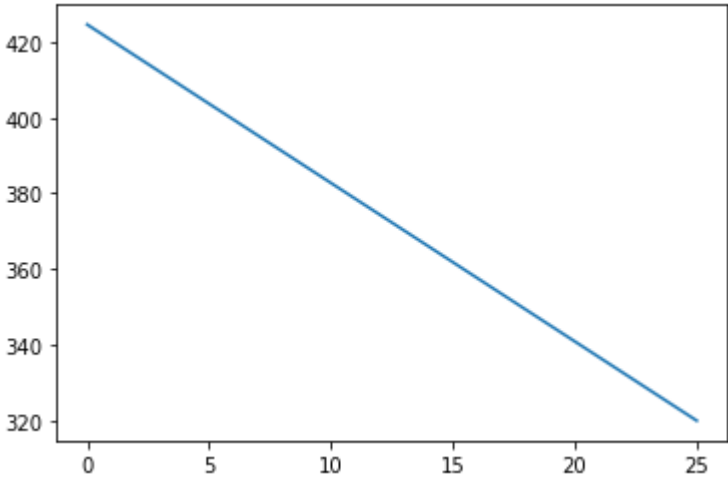
#utilizamos Ufuncs te numpy para hacer la Ecuacion General de la Regresion

y= intercept + coefficient * x
y #Resultado
```

Out[7]: array([424.58280657, 420.39823009, 416.2136536 , 412.02907712, 407.84450063, 403.65992415, 399.47534766, 395.29077118, 391.10619469, 386.9216182 , 382.73704172, 378.55246523, 374.36788875, 370.18331226, 365.99873578, 361.81415929, 357.62958281, 353.44500632, 349.26042984, 345.07585335, 340.89127686, 336.70670038, 332.52212389, 328.33754741, 324.15297092, 319.96839444])

```
In [8]: #creamos una Grafica con el Resultado Obtenido
plt.plot(y)
```

Out[8]: <matplotlib.lines.Line2D at 0x1bc428b3af0>



```
In [ ]:
```