Linear Regression On MNIST

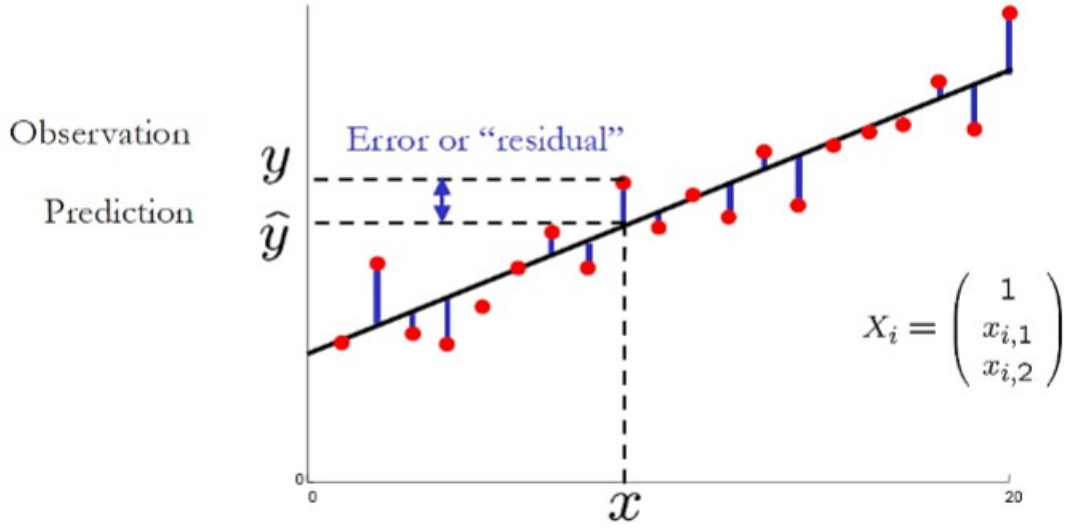
Homework 3- Hector Ramirez
Python Version: Python3
Libraries used: Pandas, NumPy, Matplotlib.pylot, sklearn

What it is:
In statistics, linear regression is a linear approach for emonstrating the connection among a scalar response at at least on logival factors (likewise known as a dependent and free factors).

In direct relapse, the connections are demonstrated using linear indicator functions whose obscure model parameters are estimated from the data. Such models are called linear models

How it does it:
The overall thought with straight relapse is that given models (xi, yi)i= 1… n foresee the yn+1 given another point xn+1. To address the information as a vector/grid, we expect the model to be y = b0 + bX + ε, where b0 and b are block and incline, known as coefficients or boundaries, ε is the mistake term, commonly expects to be that ε ~N(μ, σ-2)

Find the ideal coefficient vector b that mentions the most comparative observable fact by limiting the amount of the squares of the contrasts between the noticed reactions and the precited straight capacity



To limit the blunder (and acquire the ideal arrangement of boundaries b) subsidiaries of the expense with every boundary should be zero

$$min\, J(b) = \sum_{i=1}^{n} (yi - xi * b)^2$$

To apply straight relapse for grouping (paired), encode class marks as y = {0,1} or {-1, 1}. Apply OLS, check with class the expectation is nearer to (client edge). Be that as it may, straight relapse isn't enhanced for grouping.

Applications:

1. Concentrating on motor execution from test information in cars

2. Least squares relapse is utilized to display causal connections between boundaries in natural frameworks

3. OLS relapse can be utilized in climate information investigation

4. Direct relapse can be utilized in statistical surveying studies and client study results examination

Results:

Experiment	TPR	FPR	Accuracy
1	0.77	0.10	85.29
2	0.89	0.00	94.12
3	0.77	0.00	85.29
4	1.00	0.28	79.41
5	0.93	0.33	78.79
6	1.00	0.00	100.00
7	1.00	0.10	93.94
8	0.86	0.09	87.88
9	0.94	0.06	93.94
10	0.81	0.06	87.88

Average Accuracy: 88.65, Average TPR: 0.897, Average FPR: 0.102

```
❯ python linear_regression.py
TPRs: [0.77 0.89 0.77 1.   0.93 1.   1.   0.86 0.94 0.81]


FPRs: [0.1  0.   0.   0.28 0.33 0.   0.1  0.09 0.06 0.06]


Accuracies: [ 85.29  94.12  85.29  79.41  78.79 100.    93.94  87.88  93.94  87.88]


Average TPR: 0.897, Average FPR: 0.10200000000000001

Average Accuracy: 88.65400000000001
```