Efecan Kasapoğlu
31038
CS412
HW2
Colab Notebook Link:
https://colab.research.google.com/drive/1d2wmOlLa3D2s4al9Tg4BEhiAbea_Mkg9?usp=sharing

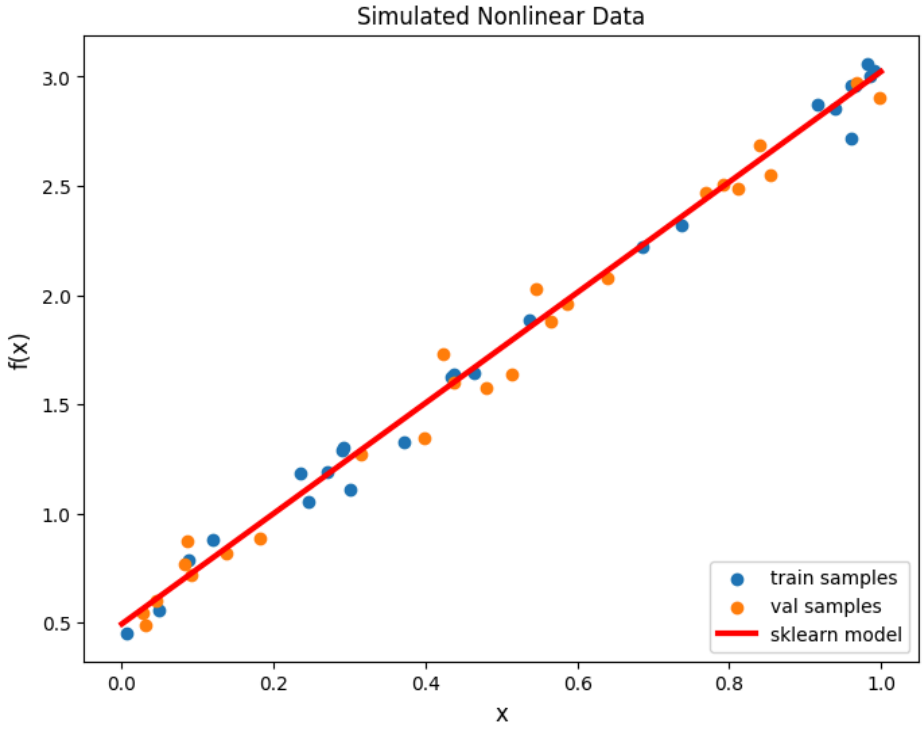
I. Introduction

In this project, I worked on linear and polynomial regression methods. Regression analysis is an important technique used to model the relationship between variables. I worked on datasets using Sklearn's ready-made models, pseudo-inverse method and gradient descent optimization. I evaluated the performance of the models with the Mean Square Error (MSE) metric. I also analyzed the effect of overfitting, underfitting and polynomial degree by visualizing regression lines and error curves.

II. Part 1

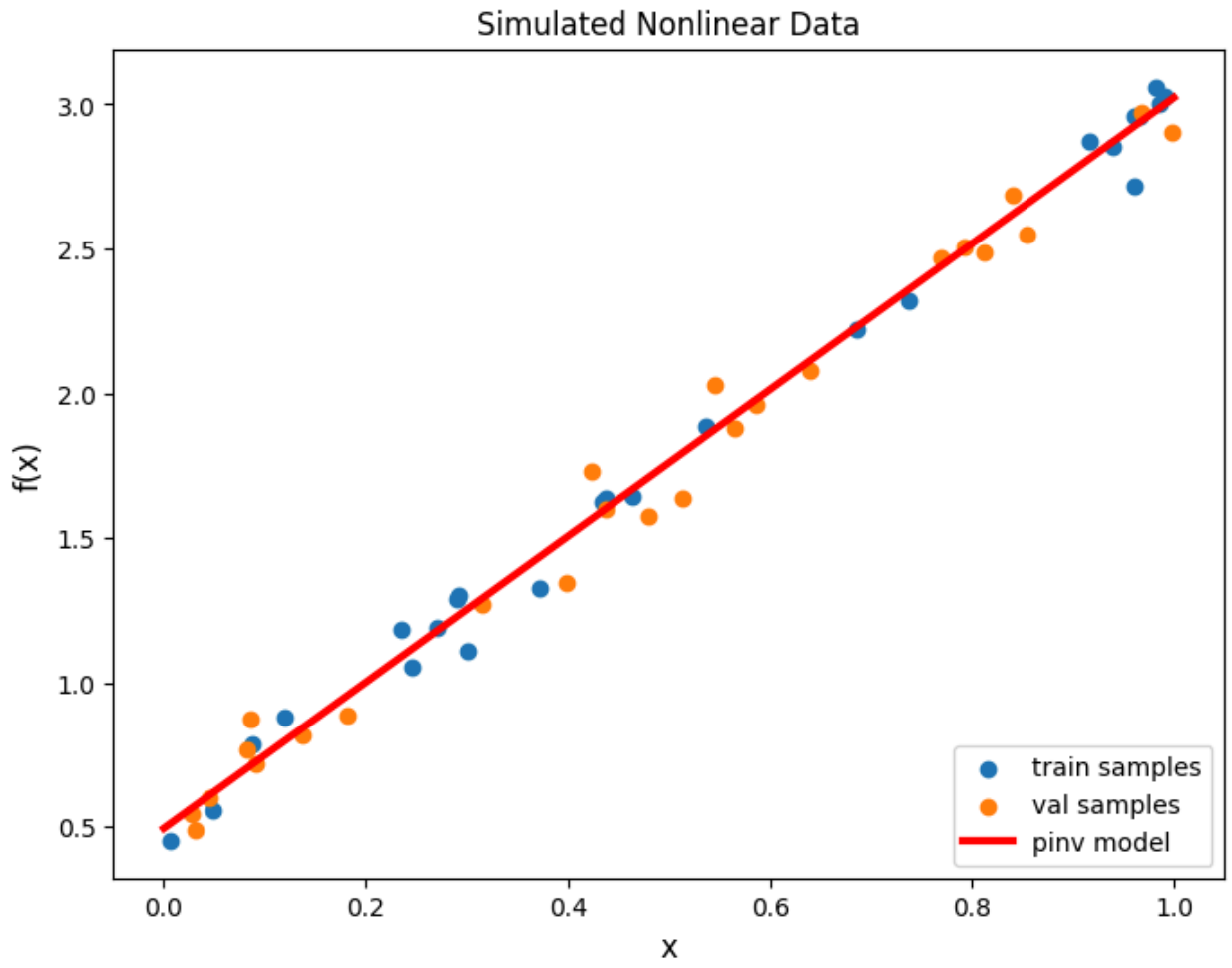
1.A Using Sklearn's Linear Regression Model

In this part I have implemented Sklearn's Linear Regression model on Dataset 1. My aim here is to train the model, make predictions on the validation set and evaluate the performance with Mean Squared Error (MSE). As a result; Mean Square Error (MSE) value of the model is **0.003977313413895165**. Low MSE value here shows a good fit. Regression coefficients w_0=0.4944766910978471 and w_1=2.5283826233685383.



1.B Using The Pseudo-Inverse Solution

In this part I worked on a pseudo-inverse solution manually to find the optimal regression coefficients. This computes the optimal regression coefficients by solving the least squares problem. As a result; MSE of Pseudo-Inverse Linear Regression is **0.003977.** The MSE value I found in this part confirms the MSE value in the previous part. It provides a good fit of the distribution of the regression lines and validates the implementation. Regression coefficients are w_0=0.49447669109784775 and w_1=2.528382623368539.

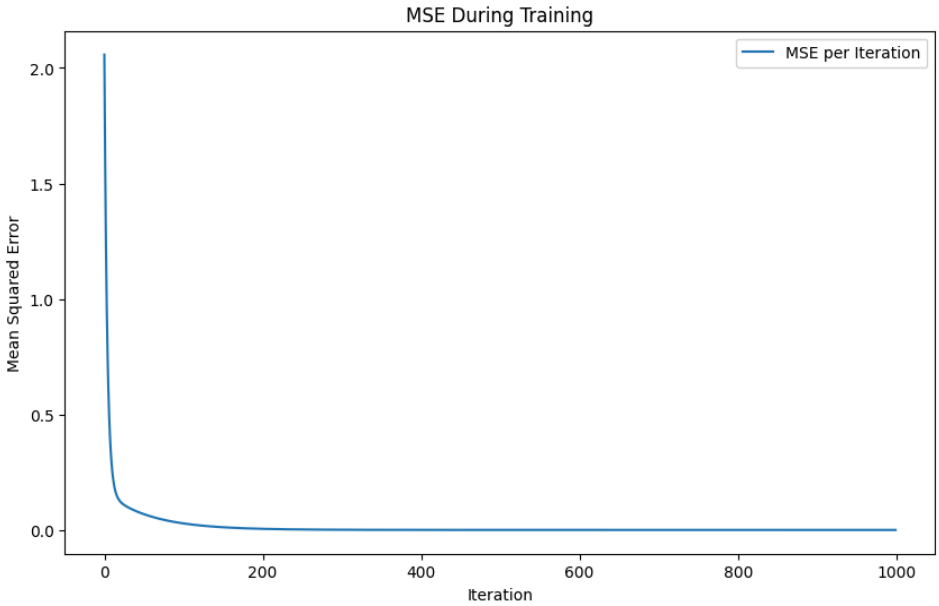


1.C Using Gradient Descent for Linear Regression

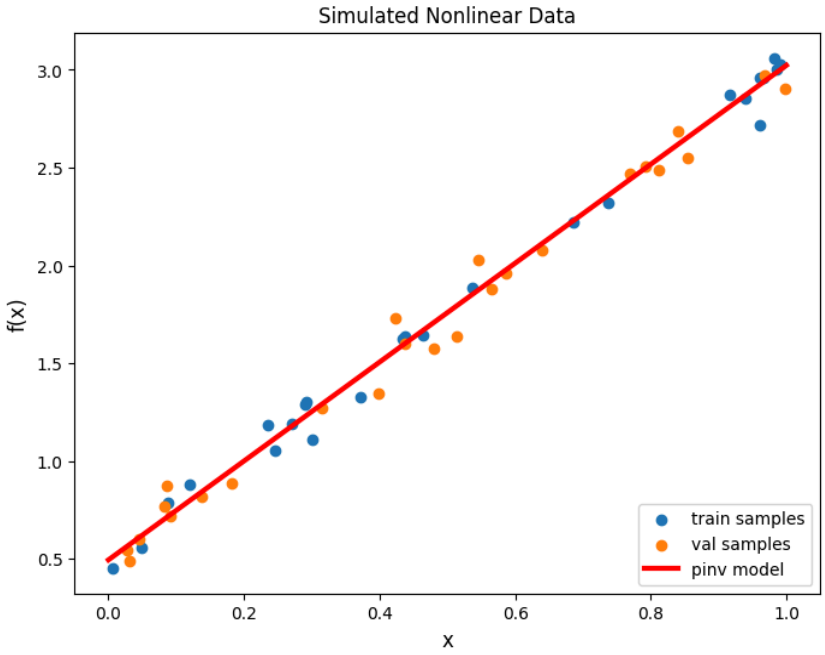
In this part I used Gradient Descent to optimize the regression coefficients for linear regression. This model is different from the 1.A and 1.B because Gradient Descent minimizes MSE by updating coefficients by step by step. After 1000 steps of iteration MSE is **0.0026**. In the **Graph A** you can see that MSE decreases and then it stabilizes, this shows successful convergence. In the **Graph B** the regression line is similar to the 1.A and 1.B graphs, confirming that Gradient Descent successfully optimizes the coefficients. Regression coefficients are after 1000 iterations are w_0=0.4946148767911626 and w_1=2.5281451790071303.

```
MSE error at step 1: 2.0573
MSE error at step 100: 0.0313
MSE error at step 200: 0.0075
MSE error at step 300: 0.0035
MSE error at step 400: 0.0028
MSE error at step 500: 0.0027
MSE error at step 600: 0.0026
MSE error at step 700: 0.0026
MSE error at step 800: 0.0026
MSE error at step 900: 0.0026
MSE error at step 1000: 0.0026
```

Graph A:



Graph B:

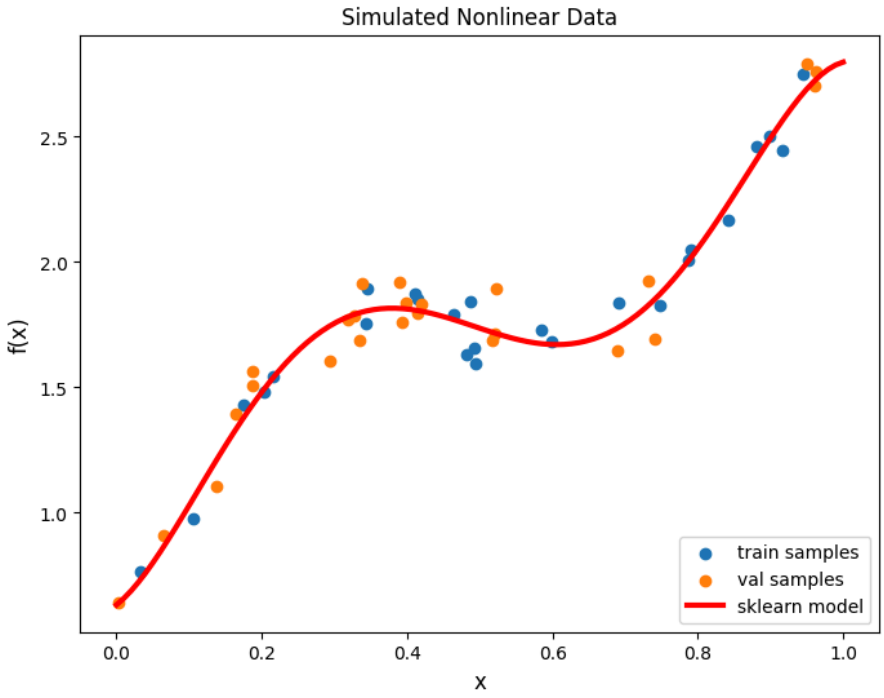


III. Part 2

2.A Performing Linear Regression On Polynomial Features

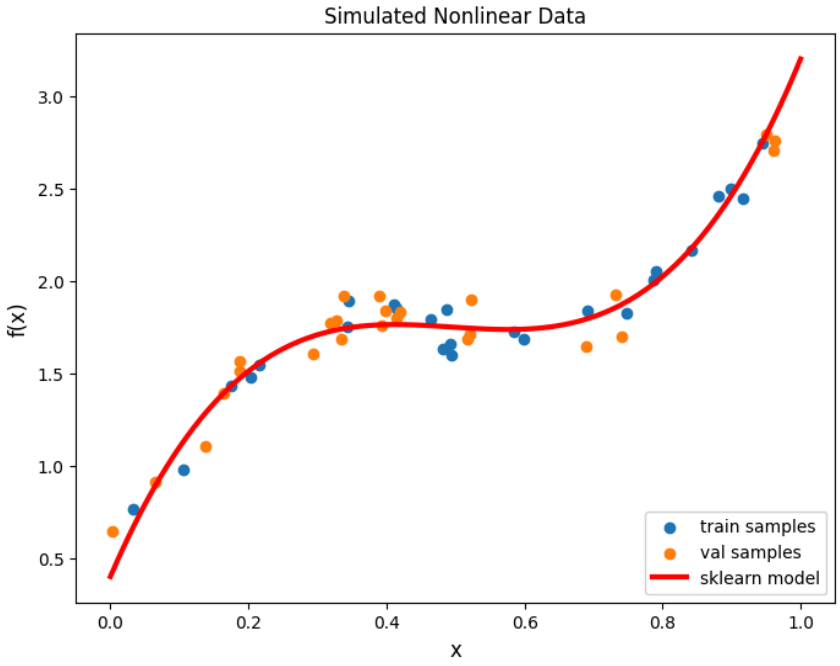
In this part I have worked with Dataset 2 which has a non-linear relationship compared with Dataset 1. I used polynomial features and used Sklearn's Linear Regression to fit the model. I have tested polynomial degrees of 1,3,5 and 7. For degree 1 MSE is **0.0318**, degree 3 MSE is **0.0060**, degree 5 MSE is **0.0037** and degree 7 MSE is **0.0058**. As a result; best polynomial degree is **5** with **0.0037**. Regression coefficients are w_o=0.63284179 , w_1=2.30261338, w_2=25.94068538, w_3=-105.99696315, w_4=135.01003494 and w_5= -55.09169475

```
Polynomial Degree: 1
MSE of sklearn model (Degree 1): 0.0318

Polynomial Degree: 3
MSE of sklearn model (Degree 3): 0.0060

Polynomial Degree: 5
MSE of sklearn model (Degree 5): 0.0037

Polynomial Degree: 7
MSE of sklearn model (Degree 7): 0.0058

Best polynomial degree based on MSE: 5 (MSE: 0.0037)
```



2.B Implement Our Own Polynomial Regression

In this part I used polynomial regression manually instead of Sklearn's functions. The MSE of the manual polynomial regression is degree **3** and the value is **0.0060**. The model performs the non-linear better than the simple linear model. However; the value of MSE(0.0060) is not the best choice compared with the sklearn's model for degree 3 (0.0037). Regression coefficients are w_0=0.39883796, w_1=8.68921502, w_2=-18.07027007 and w_3=12.18401084.



IV. Conclusion

For Part 1, Sklearn's Linear Regression model and The Pseudo-Inverse Solution gives similar results. However; Gradient Descent for Linear Regression needs iterations to give optimal results and this iteration step takes more time compared with other models. However; after iterations the Gradient Descent model gives similar results with other models. For part 2, for degree 1 underfitting happened due to the linear model not flexible enough. On the other side; for degree 7 the value of MSE increased due to overfit. That is why degree 5 is the best result. It is clear that selecting the right polynomial degree is crucial for overfitting and underfitting.