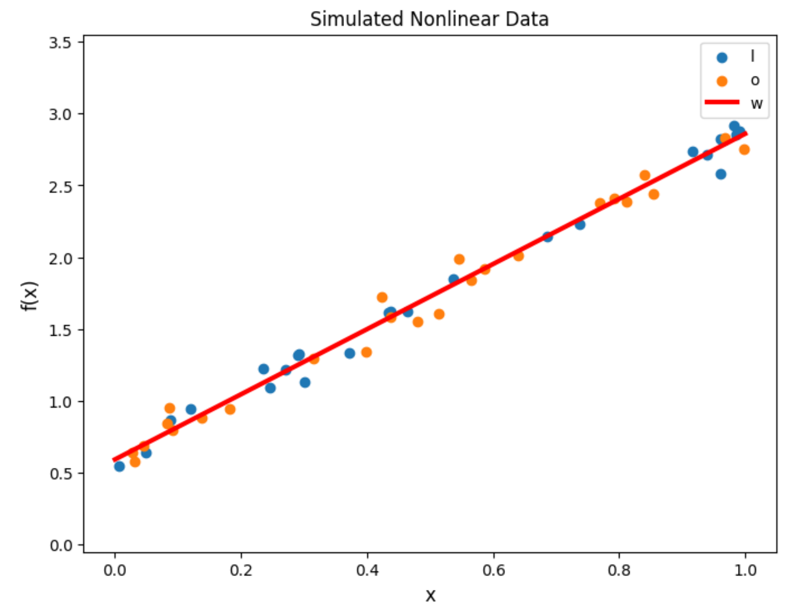
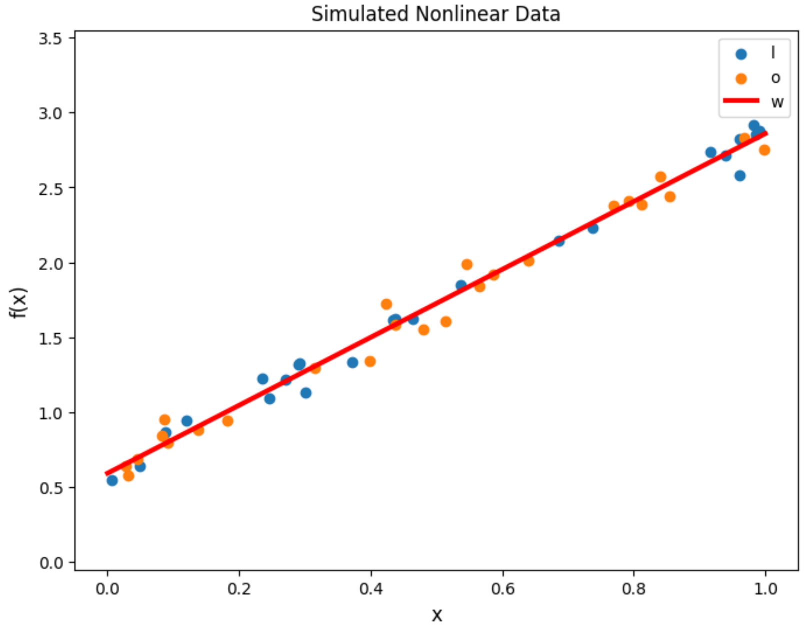
https://colab.research.google.com/drive/1vmHhnNHc7y2OU3pRTZCF0_FyRQZYMaSj?usp=sharing

Your report should contain sections in the same order as you're seeing in this notebook and labeled as Part 1.a Results etc). In each part, you should include the regression coefficients you have found in that part and all the plots and MSE errors.

Part 1.a

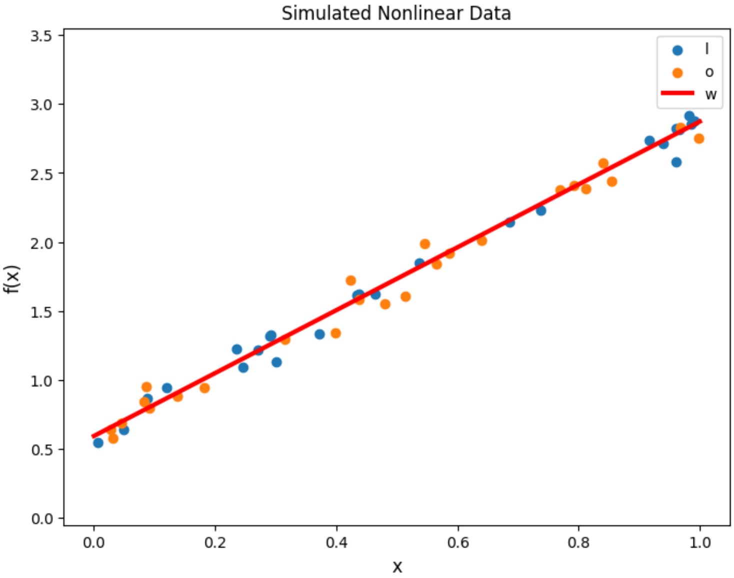
MSE of sklearn model: : 0.007735940505474359
Regression coefficients: [[2.26617099]]

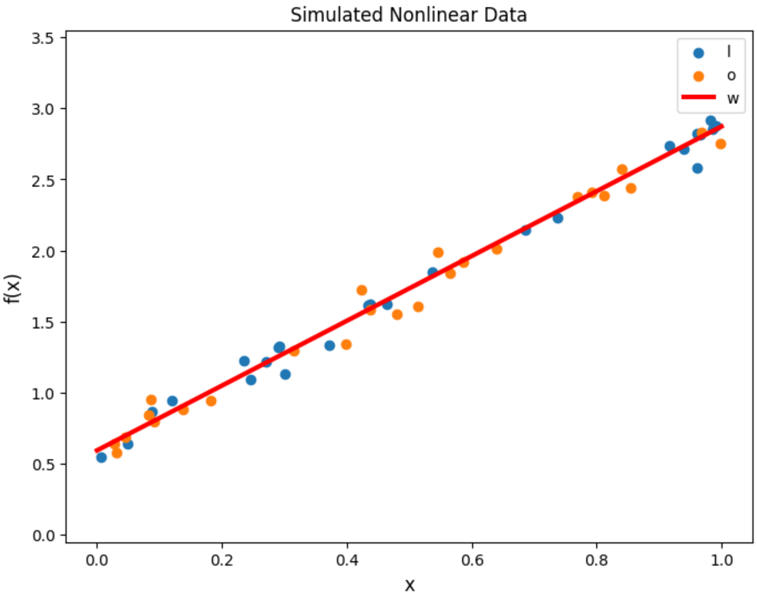




Part 1.b

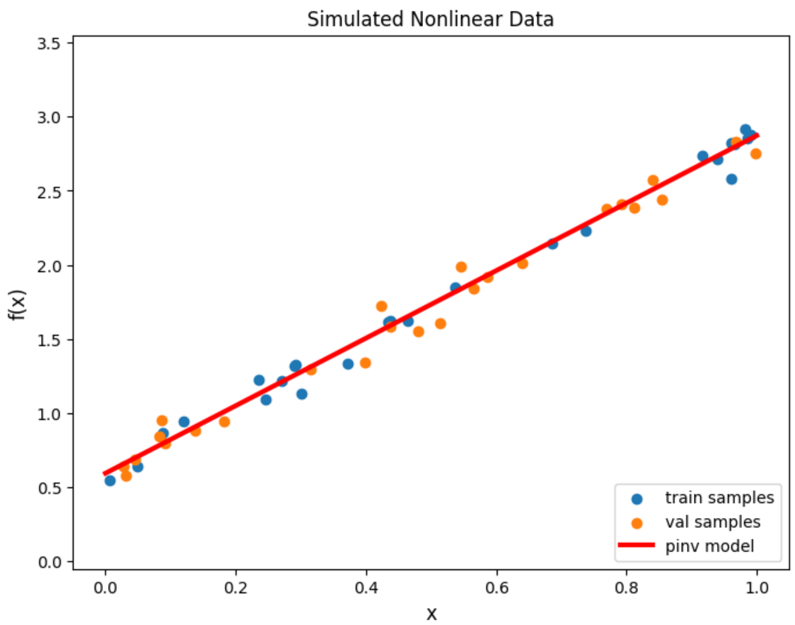
MSE of manual model: 0.007954626827790367
Regression coefficients: [[0.59447669][2.27838262]]

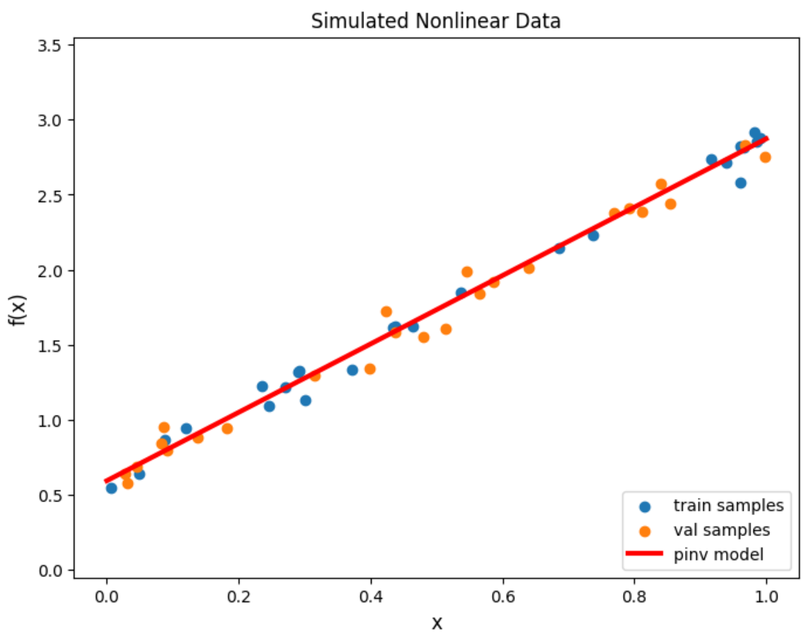




Part 1.c

MSE error at step 1: train: 1.9355, val: 1.3955
MSE error at step 100: train: 0.2045, val: 0.1907
MSE error at step 200: train: 0.0391, val: 0.0382
MSE error at step 300: train: 0.0110, val: 0.0127
MSE error at step 400: train: 0.0063, val: 0.0086
MSE error at step 500: train: 0.0054, val: 0.0080
MSE error at step 600: train: 0.0053, val: 0.0079
MSE error at step 700: train: 0.0053, val: 0.0079
MSE error at step 800: train: 0.0053, val: 0.0079
MSE error at step 900: train: 0.0053, val: 0.0080
MSE error at step 1000: train: 0.0053, val: 0.0080
Regression coefficients: [[2.27793964] [0.5947345]]





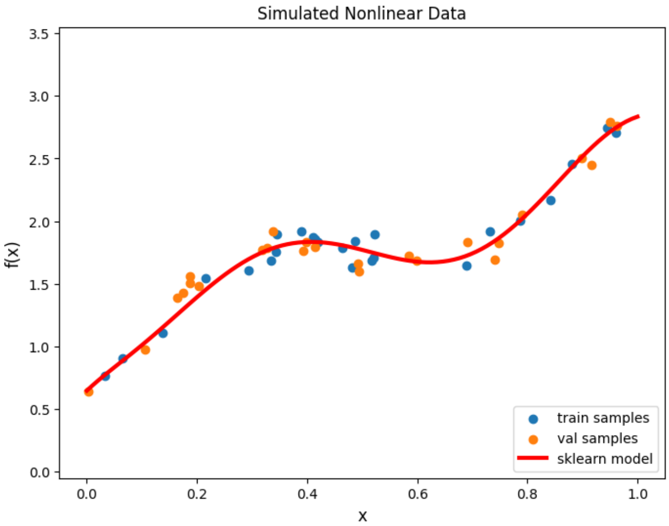
Part 2.a

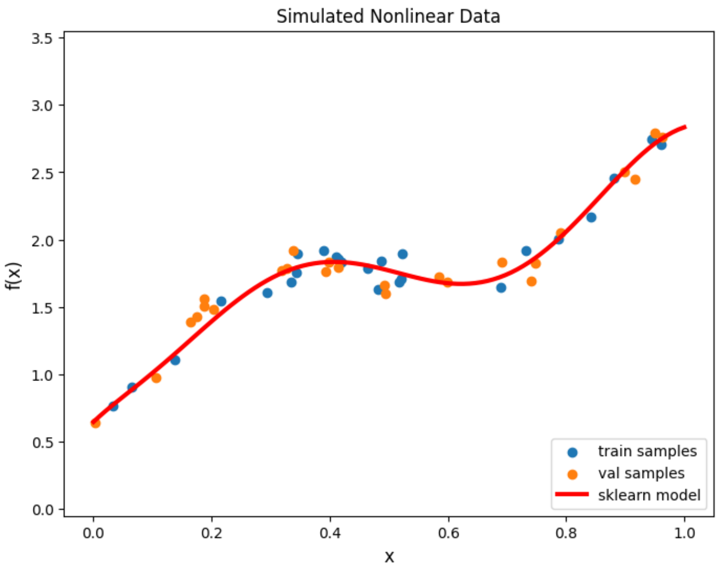
MSE of sklearn model with degree 1: 0.007954626827779035
Regression coefficients of sklearn model with degree 1: [[0.   2.27838262]]

MSE of sklearn model with degree 3: 0.008071315659914082
Regression coefficients of sklearn model with degree 3: [[0.   2.62427105 -0.93856873 0.63315782]]

MSE of sklearn model with degree 5: 0.008137198861358443
Regression coefficients of sklearn model with degree 5: [[0.   5.00078794 -17.10671311 42.32515206 -45
.35418158
   17.55680103]]

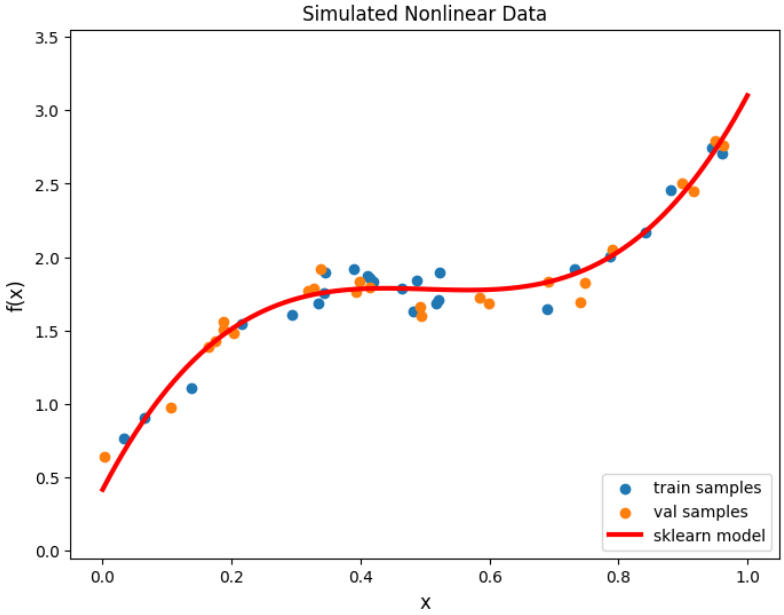
MSE of sklearn model with degree 7: 0.008922542716622659
Regression coefficients of sklearn model with degree 7: [[0.   4.07516041   2.50421109   -94.60854693 3
88.26837189
  -667.84413506 527.18669613 -157.19825457]]

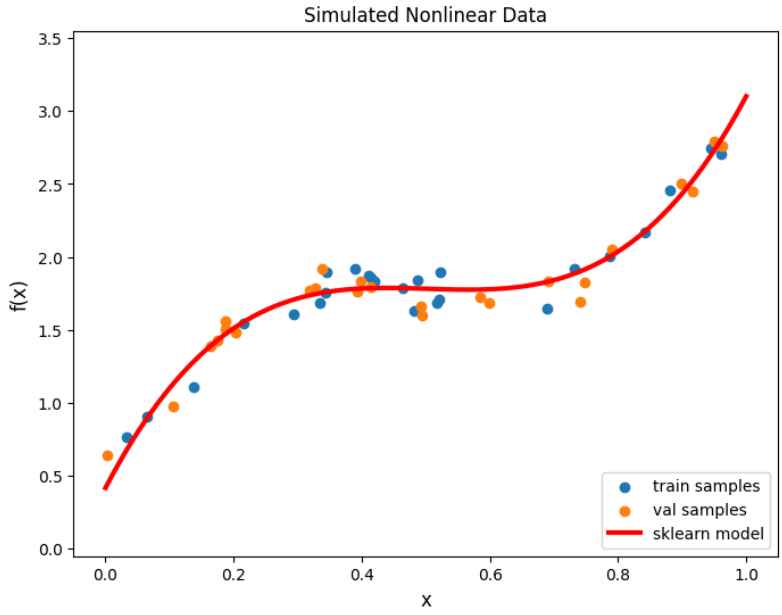




Part 2.b

MSE of sklearn model: 0.009619587973839322
Regression coefficients: [[0.41769447][8.42034259][-17.02648148] [11.28934053]]





In Part 1, you should comment on whether the gradient descent solution is the same (or very close) to solutions obtained for Part1.a and b. If not, add a line of explanation as to why you think it is not.

Both solutions in Section 1a and 1b are aiming to solve the linear regression problem and find the optimal parameters of the linear regression model that minimize the mean squared error (MSE) on the validation set. However, the implementation details are different.
In Section 1a, the sklearn library is used, and the LinearRegression model is initialized and fitted to the training data. Then, the model's predictions are found on the validation set, and the MSE is calculated using the mean_squared_error function from sklearn.metrics.
In Section 1b, the pseudoinverse of the extended data matrix is found, and the regression coefficients are calculated using matrix multiplication. Then, the model's predictions are found on the validation set, and the MSE is calculated using the mean squared error formula.
Although both solutions are solving the same problem, they are not identical. The mathematical approach used in Section 1b is more explicit, and the implementation is done manually, whereas in Section 1a, the implementation is done using a pre-built library.
In summary, while the goal of both solutions is to minimize the MSE, the implementation details and the mathematical approach used are different. Therefore, the results obtained from both solutions may differ slightly.

In Part 2, comment on the effect of the degree **parameter. What happens when it is chosen too small or too big? What do you think is the optimal** degree **value, and why? Discuss from the perspective of underfitting/overfitting**.

Polynomial degree parameter determines the flexibility of the polynomial model, i.e., it decides which degree of polynomial will be used. When a small degree is selected, the model is simple, and there is a high risk of underfitting. Conversely, when a very high degree is selected, the model becomes too complex, and there is a high risk of overfitting. In this case, the model may fit the training data very well, but its generalization performance may be poor, resulting in low performance on test data.

The optimal degree value may vary depending on the dataset and the complexity of the model. Generally, selecting a degree value between 3 and 5 yields good results. However, the model's performance may differ depending on the degree value, so it is essential to try different degree values and test the model's performance. Additionally, increasing the complexity of the model using more data may allow higher degree values to be used.From an underfitting perspective, when a very small degree is selected, the model cannot explain the data well enough, and the performance is low.

From an overfitting perspective, when a very high degree is selected, the model explains the data very well, but the generalization performance is poor. Therefore, it is important to choose an appropriate degree and test the model's performance on both training and test data.