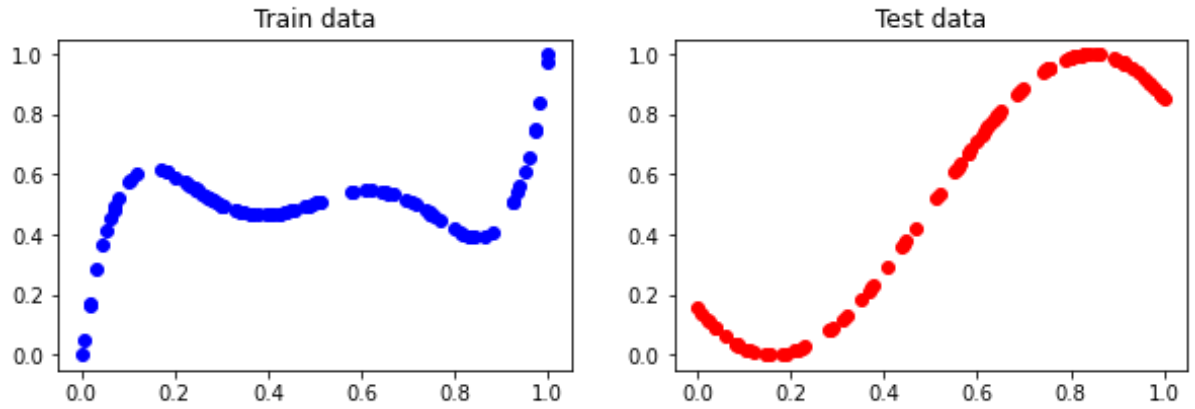
GitHub: https://github.com/atik666/CS5783/tree/assignment1

Question 1

1. Try to plot this relationship on your own using matplotlib. You can also visualize the test data
to see if it gives you any clues about the underlying relationship between the variables.



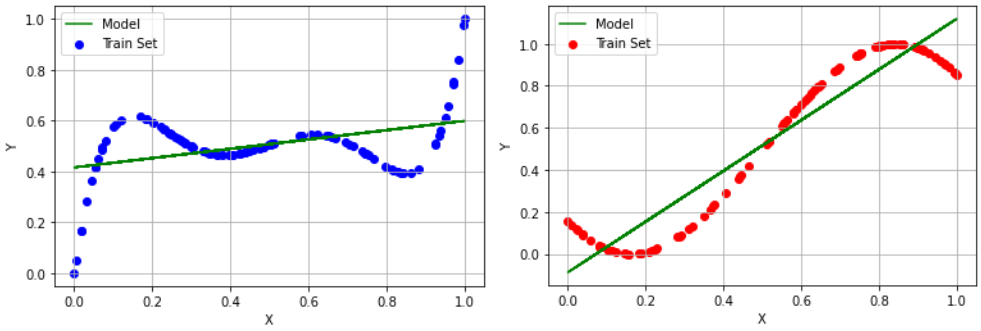
The visualization after normalizing the train and test data shows that the samples are not linear, and it is likely that they need a polynomial function to draw the boundary line.

2. Use your knowledge gleaned from the previous step to answer the following questions:

a. Is the relationship linear?

The relationship is non-linear, as appears in the graph. They're it requires a higher order polynomial function to solve it.

Here is what happens when we try to fit a linear boundary through the curve.



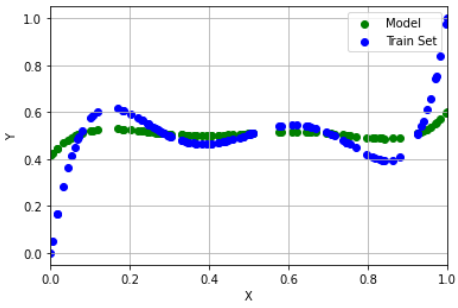
b. Do you need feature engineering to add any non-linearity?

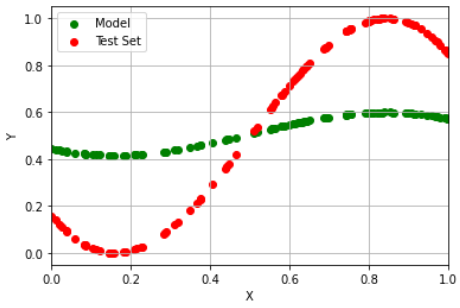
Feature engineering is needed to add non-linearity. I tried a higher-order polynomial function to add non-linearity.

I varied the order of the polynomial to find the optimal one, and for the 5th order of the polynomial degree, I achieved the lowest loss.

2nd order:

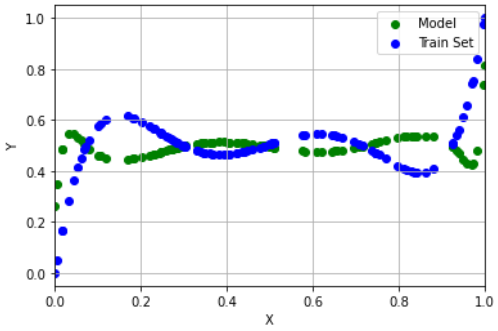
Lowest train loss = 0.0144

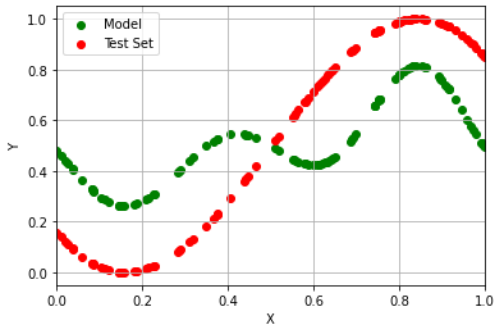




5th order:

Lowest loss = 0.0071.





[Question 2]

1. What is the average least squares error for the given data using your simple linear regression model?

I trained with 21 rows of the data and tested with 7 rows.

My lowest train loss I got was 1.064 for my last gradient descent.

Using the test data, the average least squares error was 0.059.

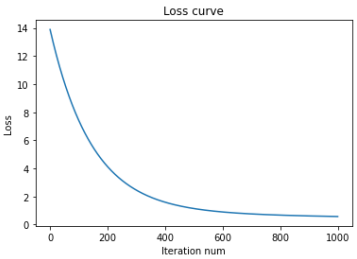
2. Which factor has the most effect on the final value? How do you know this? Can you use only
this feature to predict the price?

The ‘Living area’ has the most effect on the final value. First, I assume that from looking at the standard deviation of this column.

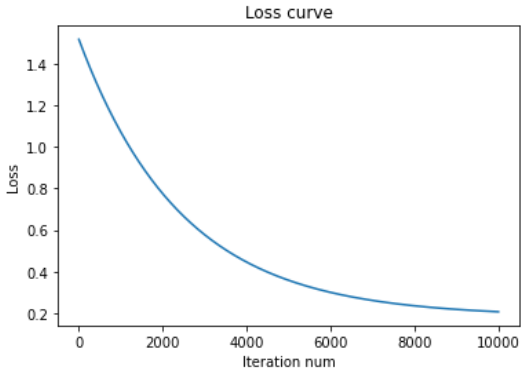
So, I trained the model without this column, and the accuracy dropped.

The average test loss was 0.0653, which was worse than the earlier.

I also tried with other columns but dropping this column had the highest test loss.



Yes, using only this value, we can predict the price. However, it much a lot more iterations to converge.

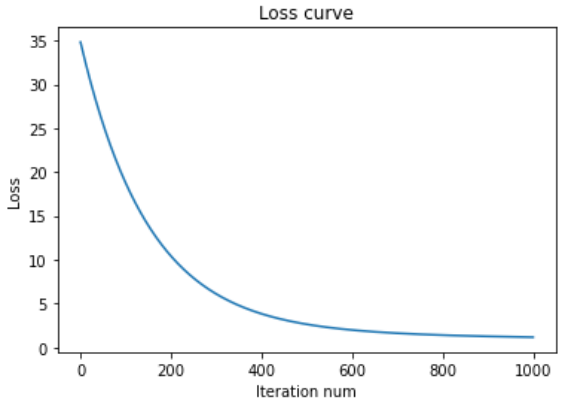


The lowest average test loss was 0.0059.

3. Which factor has the least effect on the final value? How do you know this? What effect does
removing this feature have on the performance?

The '# Fire places' column has the least effect on the final value. My guess was also from the standard deviation.

So, I trained and tested by removing this value. The test accuracy improves since it gets easier for the model to draw the boundary with fewer dimensions.



The test average loss is 0.0591, which is lower than the earlier model.

[Question 3]

1. Do you need any basis functions when using the locally weighted approach?

No, basis function is not necessary for locally weighted approach as it is non-parametric learning.

2. What is the difference between this implementation and the one for Question 1?

The primary difference from the gradient descent approach of linear regression would be that there exists no training phase. All the work is done during the testing phase/while making predictions. Also, It is a non-parametric algorithm.