**CS 5783**

**(Machine Learning)**

**Assignment 1**

Github: <https://github.com/RafaeAbdullah/CS5683-1>

**By**

**Name: Rafae Abdullah**

**Email:** [**rafae.abdullah@okstate.edu**](mailto:rafae.abdullah@okstate.edu)

1.1. The following figure shows relationship between the x and y variable from training data.

Chart, line chart

Description automatically generated

Since test data are supposed to be unseen, I am not plotting them. But this plot is enough to understand their relation.

1.2.a) No, the relationship is not linear.

1.2.b) We need feature engineering to add non-linearity. Another way to deal with this is using locally weighted approach for building regression model.

1.2.b) i) We can use basis function for feature engineering.

1.2.b) ii) Looking at the various plots of different powers of x in google and in the lecture slide (ref: lecture-3, slide-52), I think x^3 and x^5 matches with the above plot of relation between x and y. Therefore, I can try these two functions to predict y from x.

1.2.b) ii) 1. After plotting, we can observe that 3rd degree polynomial can predict the target variable y moderately for train data, but not very well for test data.

|  |  |
| --- | --- |
| Plot for training set | Plot for testing set |
|  |  |

However, when I add x^5 besides x and x^3, the result is impressive.

|  |  |
| --- | --- |
| Plot for training set | Plot for testing set |
|  |  |

We can see, the predicted y perfectly matches with original y values. As a result, least square error in both training and testing set is approximately zero.

2. At first, I split the dataset into training testing set with 70:30 ratio. Random seed was 12345.

The dataset has 28 records, 20 records in training set, and 8 in testing set. There are 13 columns including ID and target variable (price of house). So, there are 11 features to predict target variable.

2.1. The average least square error for linear regression model on training data is 3.98 (approx.), but on test data, it is 58.15 (approx.). Since the dataset is very small, this type of difference is not quite unexpected.

2.2. To know the importance of a variable, we need to calculate standardized coefficient. Using training dataset, here is the list of standardized coefficients.

Text

Description automatically generated

We can see 1st feature (2nd column of dataset), i.e. Local Price is the most important variable.

After using only this variable to build model, the average least square error for training data becomes 10.92 (approx.) from 3.98 (approx.), and for test data, it becomes 74.66 (approx.) from 58.15 (approx.).

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Thus, it will not be very wise to use only this variable since it shows a noticeable increase in the average least square error.

2.3. Again, from the above figure, we can see the 7th feature (8th column of dataset), i.e. Number of Bedrooms is the least important variable.

After removing only this variable and building model with the rest 10 features, the average least square error for training data becomes 5.34 (approx.) from 3.98 (approx.), and for test data, it becomes 54.78 (approx.) from 58.15 (approx.).

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Thus, removing this feature does not have noticeable negative impact on training data. Rather, dropping it helped to drop the average least square error in test data. We can assume it happened because this extra feature was causing overfitting. Hence, it will be wise to remove this variable and reduce model complexity.

3. Locally weighted linear regression model is another way to predict target for this kind of non-linear relationship. Here is the output for weighted linear regression model:

Chart, scatter chart

Description automatically generated

3.1. We do not need much feature engineering, such as using basis function. To use weighted linear regression model.

From the above figure, we can conclude that weighted linear regression model performs very well to predict target values without using basis function.

3.2. In weighted linear regression approach, we calculate the theta for a point based on the theta for the other values which are close to the point. To calculate theta, all of the previous data are used, and thus this is also known as non-parametric model. On the other hand, in case of the model using basis function approach, we calculate the theta and then we no longer use the training dataset again to calculate theta.