ECE 595 Machine Learning

Project 1: Linear Regression

Gilson Moh, Jr. & Hong Phuc Nguyen

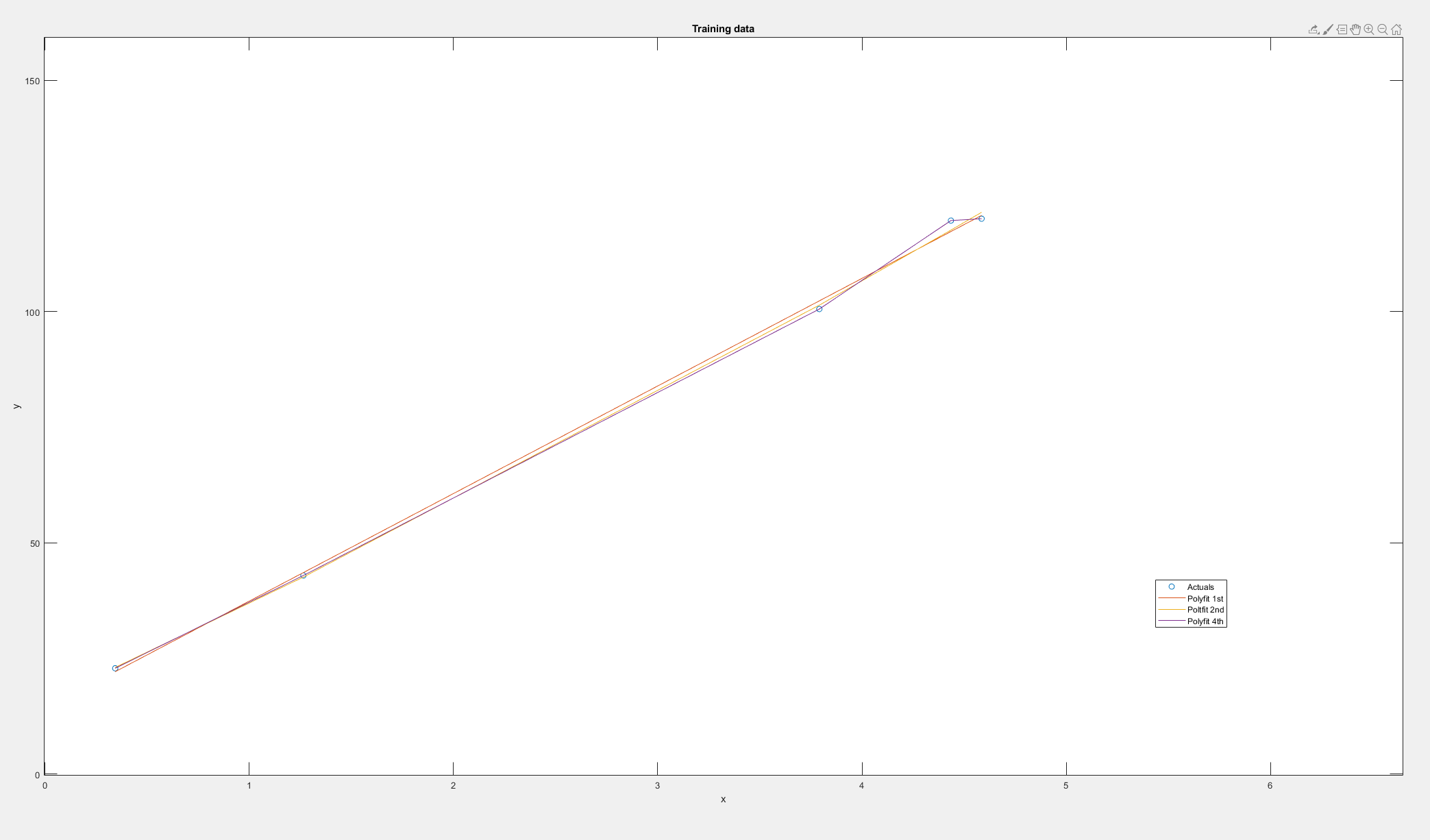
Project Due 10/3/2020

Introduction

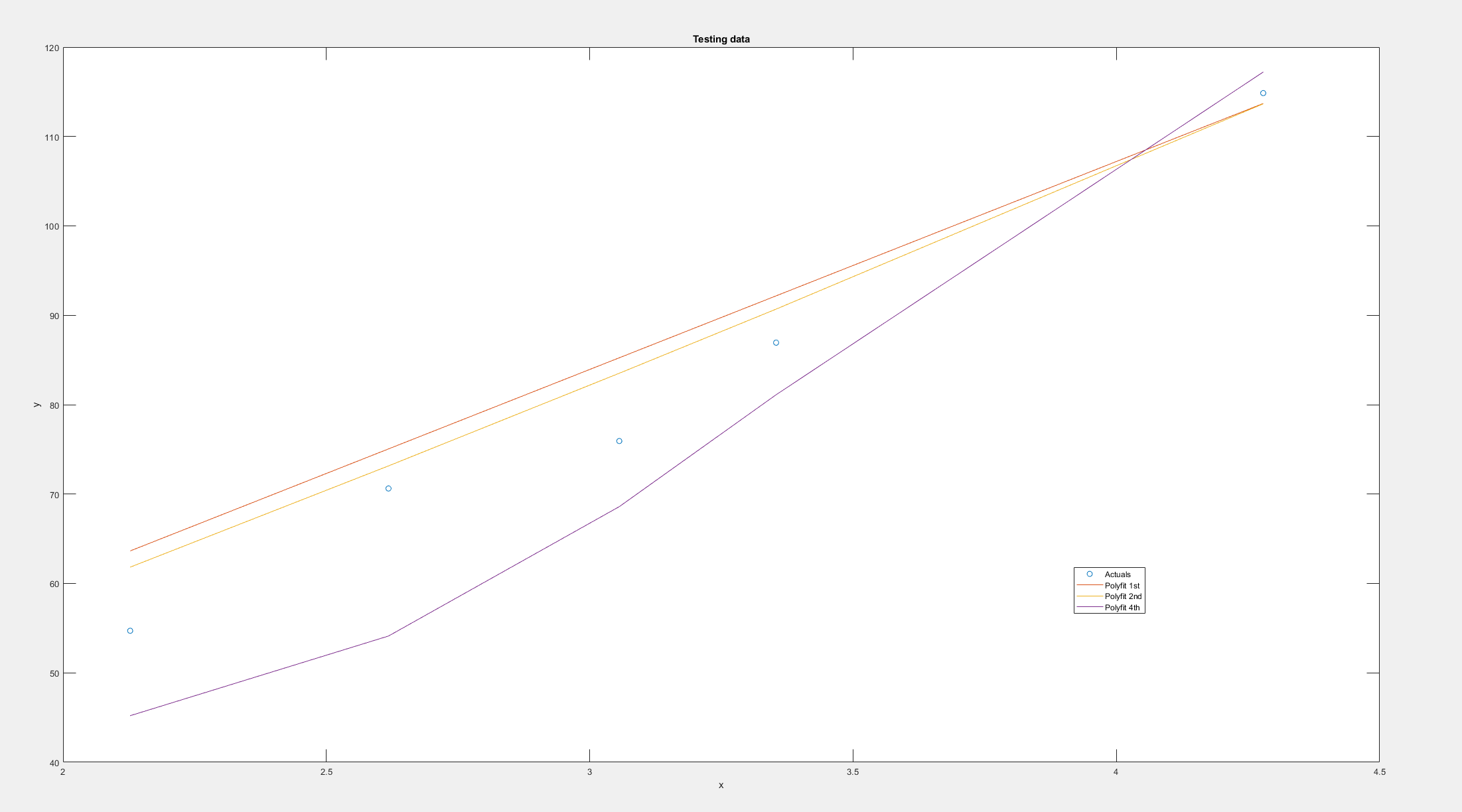
The goal of the project is to complete the two tasks given. The first task is to use a given 2nd -order polynomial equation and create a training data set. And then, perform a 1st, 2nd, and 4th order polynomial model to for the data while evaluating Root Mean Square Error at the end. The second task is to take K subfields in a data set to predict the price of a house. Three different models will be used to perform this: The first model (Model 1) is a linear combination of a constant (i.e. the number “1”) and 5 chosen parameters from the datasets (x1 to x5). The second model (Model 2) is a linear combination of all these terms, plus the squares of each parameter and all second-order cross-multiplication terms (e.g. x1x2, x1x3, etc.). The third model (Model 3) is a linear combination of all the terms in Model 1, plus the fourth power of each parameter and all fourth-order cross-multiplication terms. Root Mean Square Error will also be evaluated here as well.

Observations and Results – Task 1

The following plots are the training data polynomial fit and the testing data polynomial fit.



**Figure 1:** Data points for the training polynomial fit data



**Figure 2:** Data points for the testing polynomial fit data

The root mean square error was calculated for the training data, that had a 1st, 2nd, and 4th order polynomial fit. The same was done for the testing data.



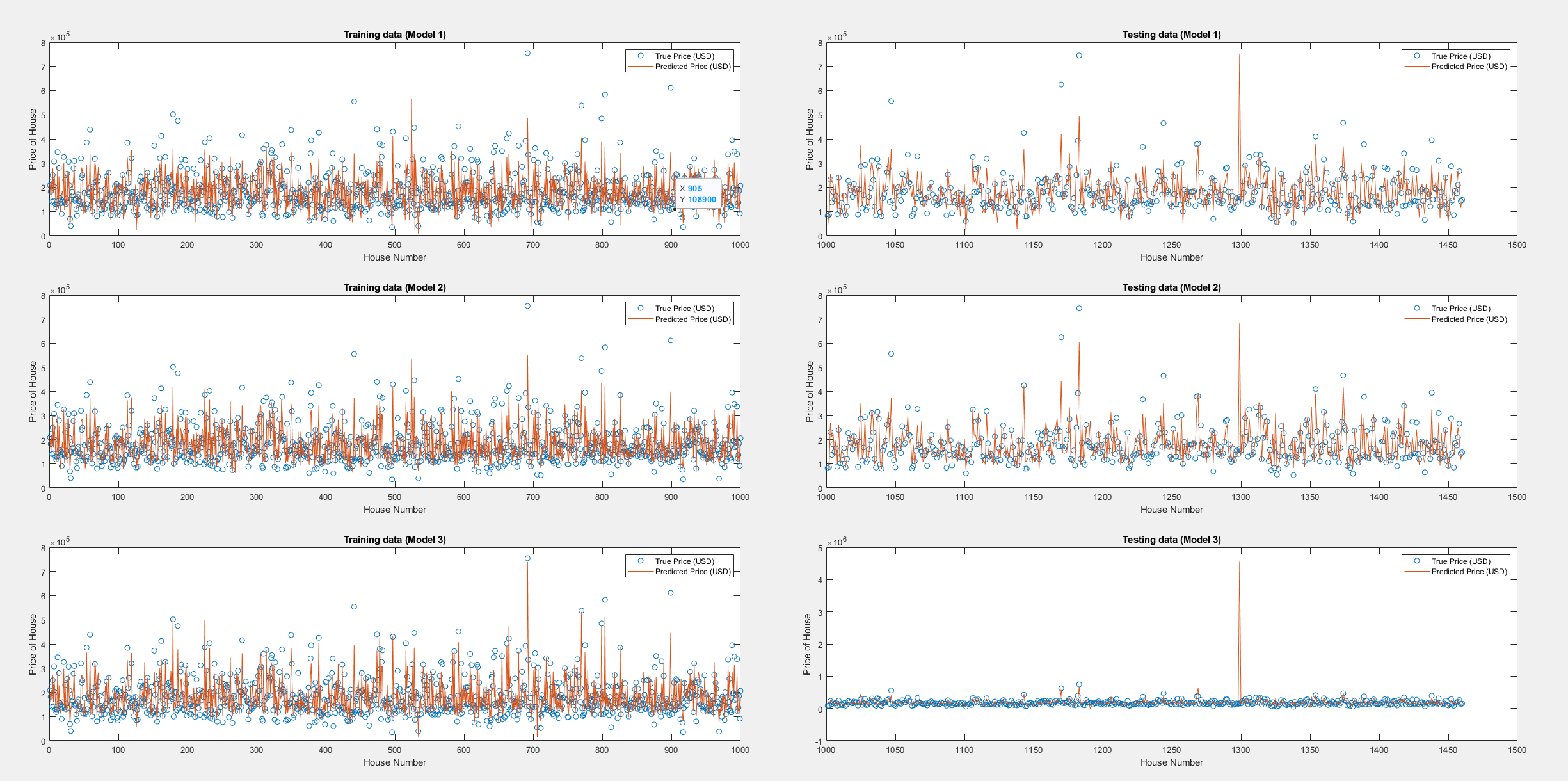
**Figure 3:** Root mean squared errors for the 1st, 2nd, and 4th order polynomial fits of the training data respectively



**Figure 4:** Root mean squared errors for the 1st, 2nd, and 4th order polynomial fits of the testing data respectively

Observations and Results – Task 2

The true price and predicated price for the training data and testing data models are plotted.



**Figure 5:** All plots for training data models and testing data models

The root mean squared error was calculated for the training and testing data models.



**Figure 6:** Root mean squared errors for the data of the training models



**Figure 7:** Root mean squared errors for the data of the testing models



**Figure 8:** Root mean squared log errors for the data of the training models



**Figure 9:** Root mean squared log errors for the data of the testing models

Discussion and Conclusions – Task 1

In conclusion, the first program was able to successfully complete task 1. The training data was created and the *polyfit* function within MATLAB was able to fit the data based on the n-th order desired. Observing the root mean squared error for the training data, it can be seen that the range is between 1 and 2.5. The second order polynomial fit had the lowest error while the fourth order polynomial fit had the highest error. The second order polynomial fit having the least error makes sense since the given equation was a second order. It can be noted that since the program used a random number generator when creating the sample data set, each time the program is run, the error changes. This is important because with a small number of elements, 5, the final results can change greatly. Thus, the second order error might not always be the smallest. As such, a greater number of elements would allow for a more proper analysis. (Informal tests using 800 data points for the training data set and 200 data points for the testing data set shows that the fact that the 2nd order polynomial fit has the minimal error always holds for sufficient numbers of data points. Moreover, the 4th order polynomial fit tends to have second-best error while the 1st order polynomial fit (linear fit) tends to have the worst error measurements, showing that as the order increases, the accuracy of the model also tends to be refined.)

Discussion and Conclusions – Task 2

In conclusion, the second program was able to successfully complete task 2. For the K-parameters, the following five parameters were chosen: 1st floor square footage, 2nd floor square footage, year built, garage space, and lot area. The program beings by extracting the desired data and formatting them in preparation for modeling. Once that data has been properly formatted, the training data and testing data are partitioned (with 1000 houses in the training set and 420 houses in the testing set). Then, the first model is trained using the training data set. Following that, the linear regression fit creates coefficients that can be used to generate a linear equation that determines the predicted house price based on the chosen parameters and the rules elaborated in the Introduction section of this report. The testing data is then inputted into the prediction equation and error is determine for both the training and testing data. The same process is repeated for Models 2 and 3. When looking at the results, for the training data, the range of the RMS error lies between $33,000 and $41,000. The higher order the model, the lower error is achieved. This may coincide with the fact that there were five parameters chosen for the prediction and each higher-order model has significantly more terms in the linear equation it produces. If a higher order model was fitted, the error might be even less. The same conclusion does not hold true for the error determined in the testing data. When looking at the plot for the testing data, it can be seen that there is a massive outlier present. The true price of the 1299th house in the data set (one such element used in the testing data set) was $160,000, but the estimated price was $4,500,000. This value is what most likely caused the error for the third model to be much higher than the first two models. If the outlier was removed or not taken into account for error, the third model would most likely have had the least error. Possible fixes might be trying to use more parameters to help determine the price of the house, or implementing rules to detect and disregard outliers. The same trend of higher order equating to less error still holds true for the first two models of the testing data set. Moving forward, RMS log error was calculated. This helped to actually downplay the effects of a large outlier. Thus, the values of the errors are significantly lower than compared to traditionally calculating RMS. The range lies between 0 and 0.3 for both the error from the training and testing data set. It can be noted that in both the training and data set RMS errors, model 2 had the least error.