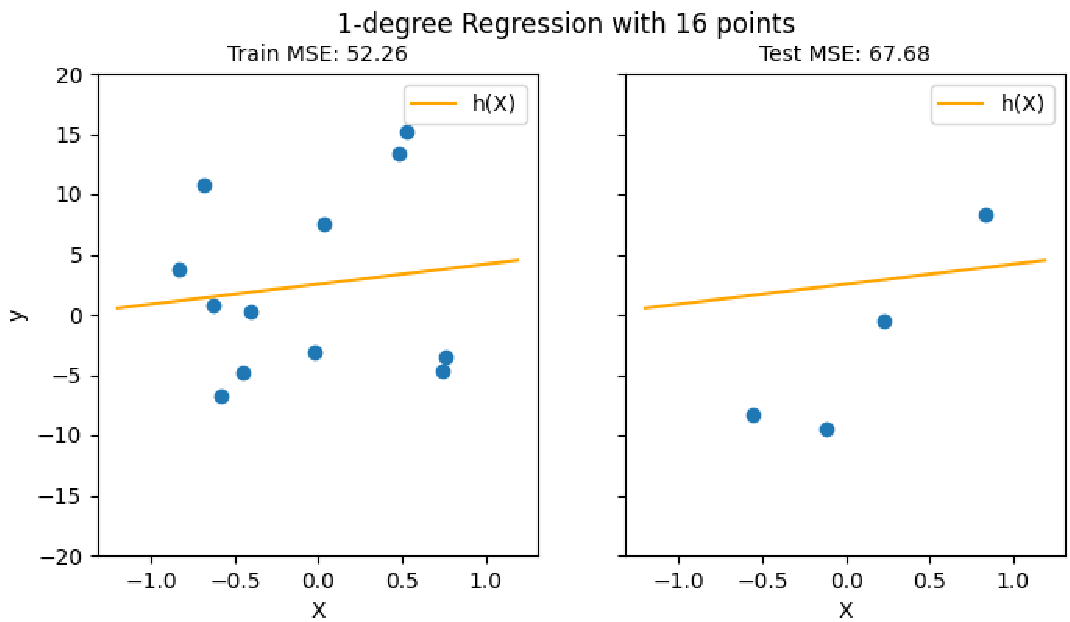
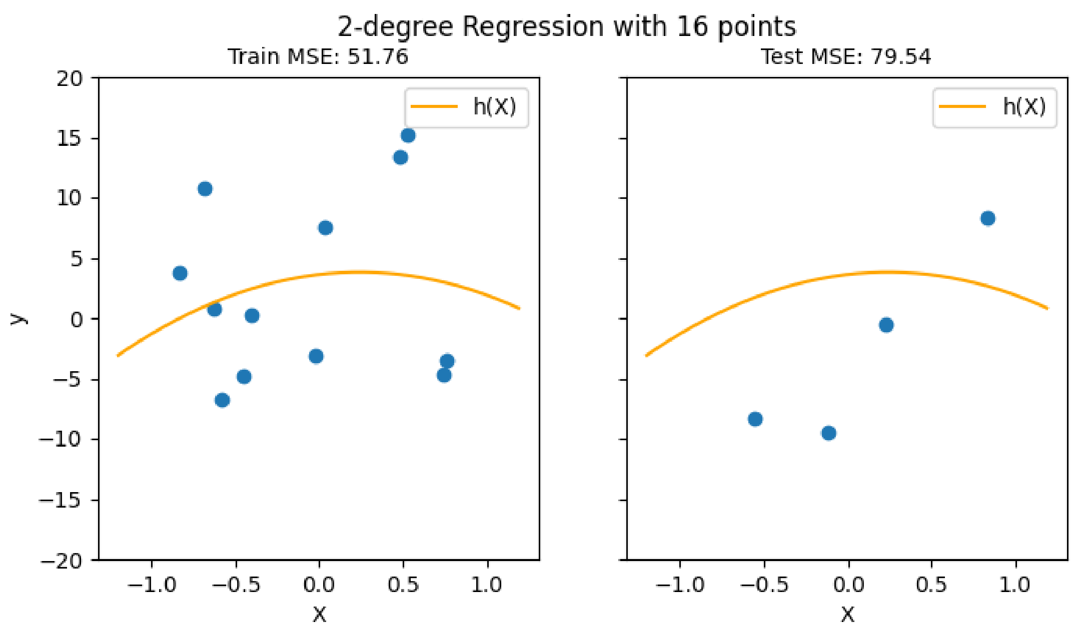
a.

Polynomial Regression

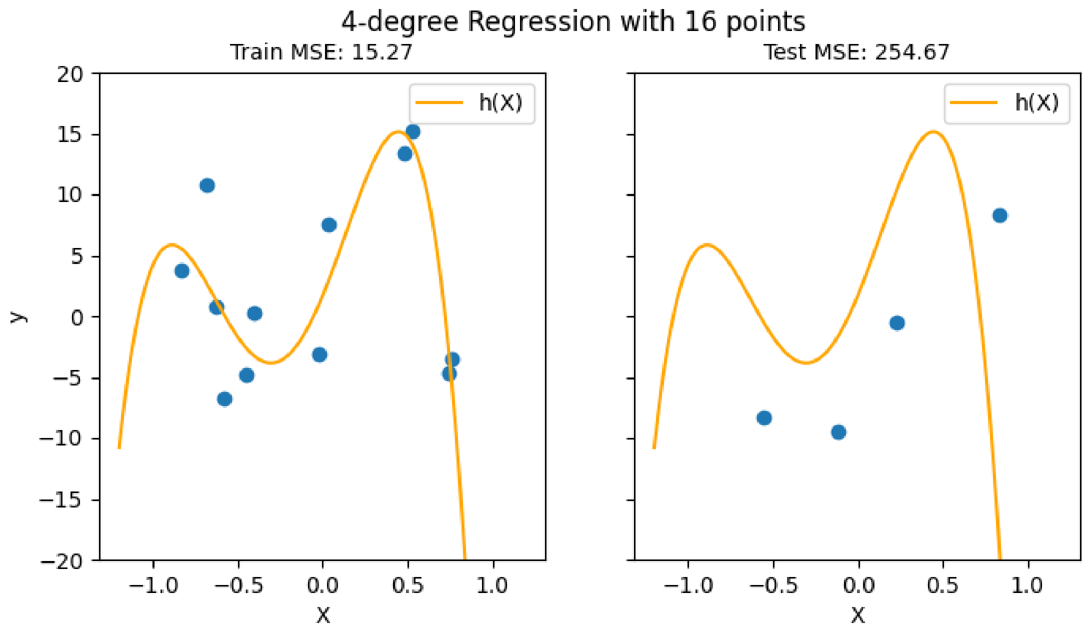
We’ll look at the relationship the degree of the polynomial regression has on the error of training the model to best fit the data with 16 points and how well the trained model translates to testing data.



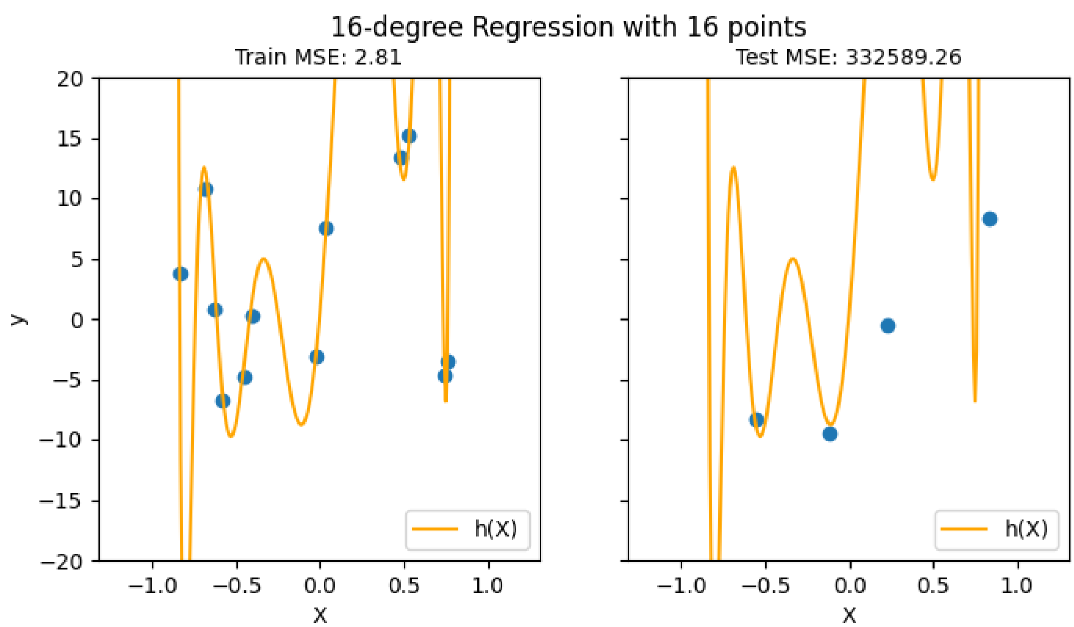
A 1-degree polynomial, linear relationship, showcases that this fit isn’t enough to capture the true relationship between the features.



A 2-degree polynomial, a parabolic relationship, showcases a very minimal slight increase in capturing a relationship between the variables as the error decreased ever so slightly, however, this presents that a 2-degree polynomial isn’t sufficient in understanding any relationship.



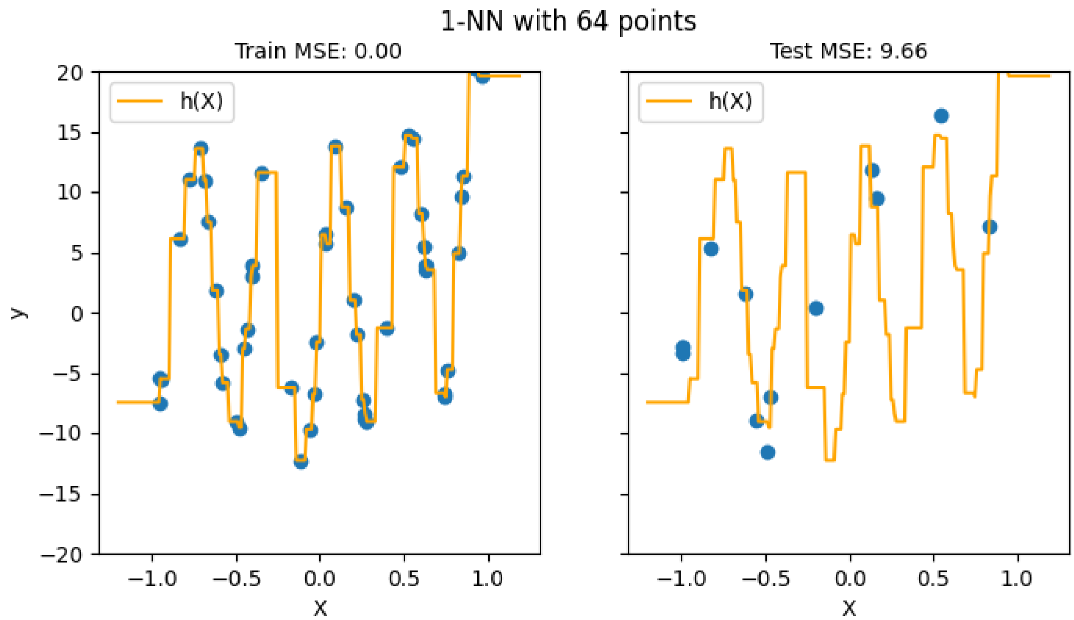
A 4-degree polynomial, showcases a pleasant decrease in error, highlighting a general relationship within the training data, however, this doesn’t translate well when applying the same model to the testing data. Though, one can see a slight trend.



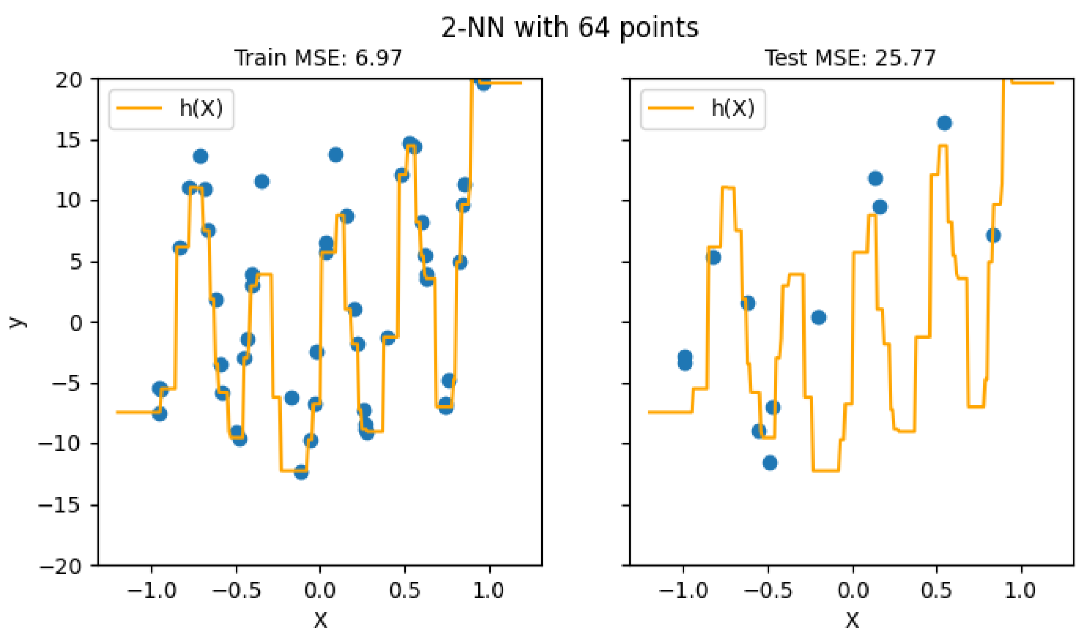
A 16-degree polynomial drastically decreases the error of the training data, which seems to be a benefit, however, applying that same model to the testing data, we obtain the highest error out of all the n-degree models. This is overfitting the training data.

K Nearest Neighbor Regression

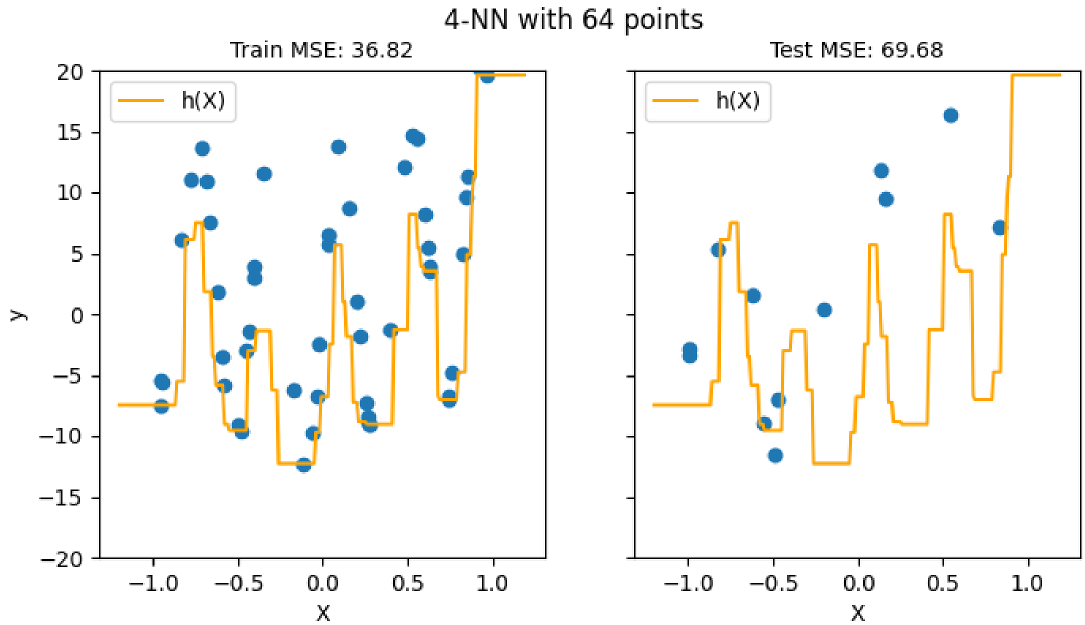
We'll look at the relationship the K-value of the K-Nearest Neighbor regression has on the error of training the model to best fit the data with 64 points and how well the trained model translates to testing data.



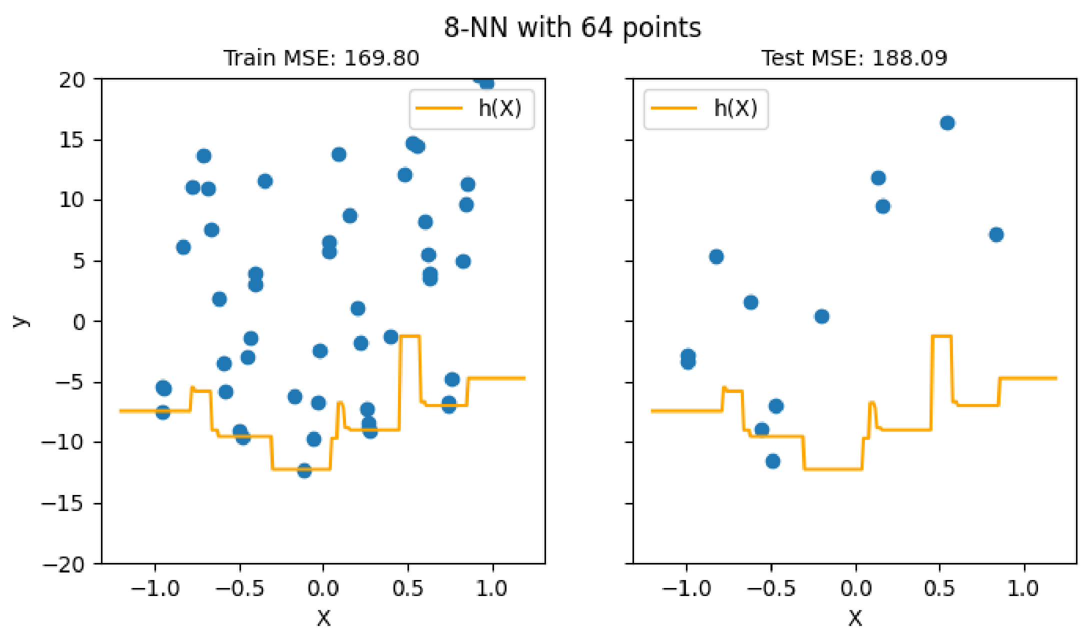
For a k-value of 1, the model swiftly fits the training data with no error. This is due to the model mapping its predicted target value to only one of the nearest neighbors, which seamlessly just traces out the flow of the data points. As we can see, this doesn't translate exactly to the training data.



For a k-value of 2, the model fits the training data fairly decently, as now we are beginning to collect more information on more neighbors for the model's predicted target value. Similarly to 1-NN, the model doesn't translate exactly with the training data but seems to have mapped out a generalized pattern found within the data.

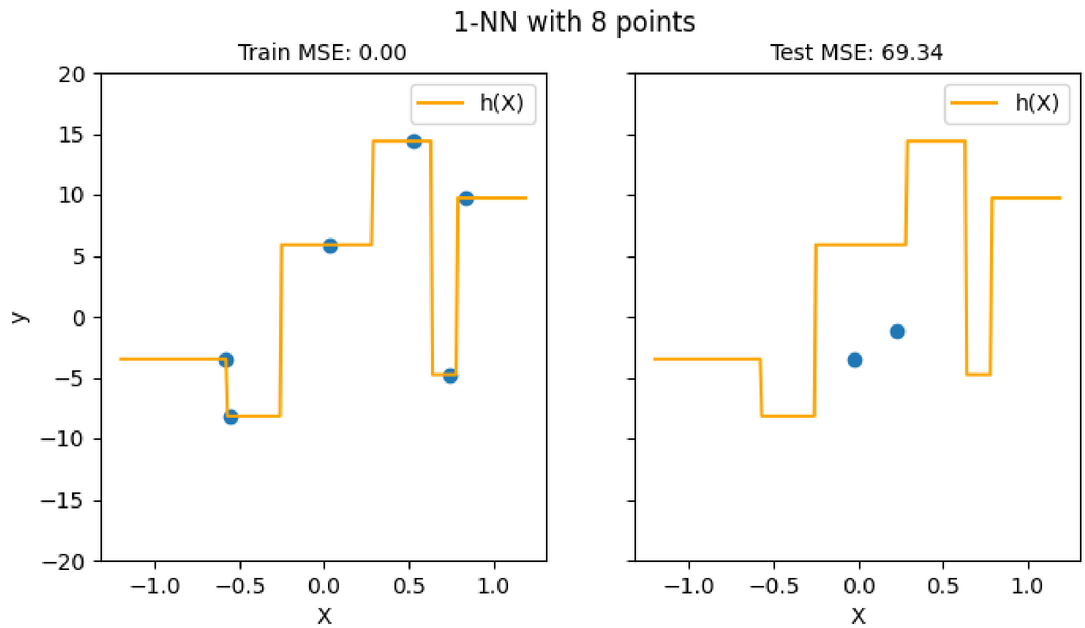


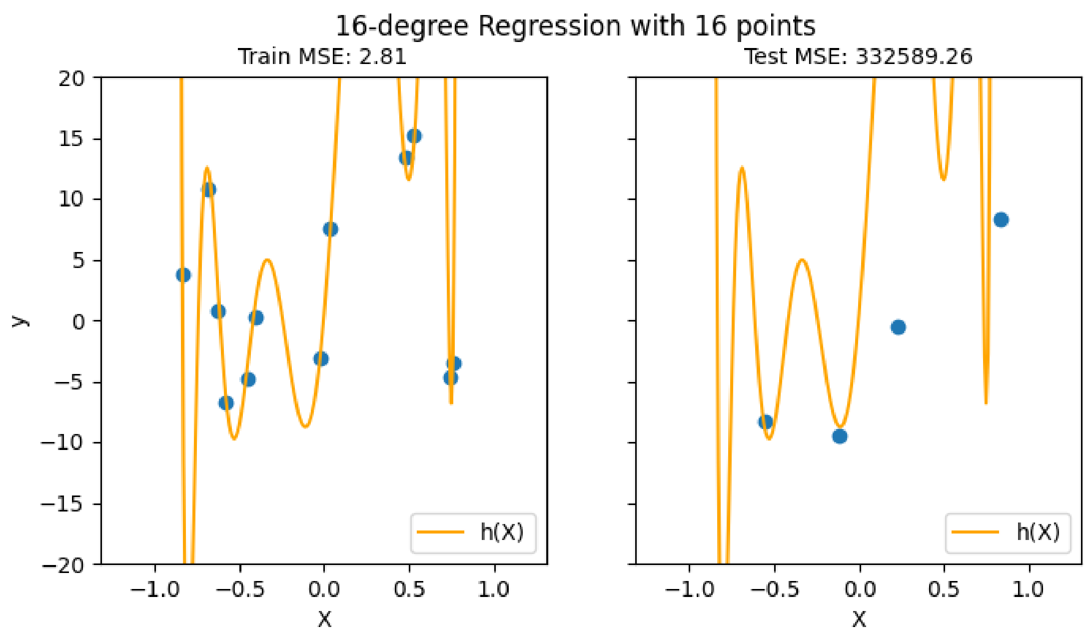
For a k-value of 4, we are now collecting the average value among 4 of the nearest neighbors for the model's predicted target value. This is beginning to show the prominent drawback of increasing how much information we ought to collect on each of the nearest data points (especially for this data set). The trained model's error becomes worse as does testing the model to new data.



For a k-value of 8, we can clearly see the effect that a large k-value has on training the model. There is simply too much information that seemingly diffuses into the average value of the 8 nearest neighbors. This shows that the model has not found a proper pattern within the data set, thus not being able to observe any particular relationship in the data.

b. Overfitting





Here, we can see two prominent examples of overfitting with our KNN and Polynomial Regression models. With KNN, having a low k-value yields a small step size between

neighbors, which will allow for the intricate mapping of how the training data behaves, yet doesn't apply to all other testing data sets.
For Polynomial Regression, having a high degree for the polynomial allows for the potential to pass through each point within the data set, however, the general flow and behavior of the data are not captured and do not apply well to the training data.

c.

Overfitting and underfitting are given light due to the relationship between the model hypothesis class and the true function that is generated from the data. In essence, overfitting occurs when a model is too complex and too specific towards a specific training data set, however, it allows for flexibility within the training data set. If the true function f(x) has noise within the training data set, the overfitting model will fit the noise. If there is a large sample set, n, the overfitting model will accommodate all points within the data set, rather than capturing the generalized behavior of the relationships found within the data. The model becomes too specific to be applied to any other testing data set as it forms only the training data set. Overfitting comes from high-degree, very small step-size models that capture the specific aspects of a training set.
For underfitting, the model is too simple and simply rigid to capture any particular behavior or relationship found within the data set. An under-fitted function may capture the very general behavior of the data set, but doesn't account for the defining characteristics of the relationships in the data set. Underfitting comes from low-degree, very large step-size models that miss the defining characteristics of the relationships in a dataset.
The interplay between f(X), H, and 'n' determines how well a model is fitting the data. The nature of f(X) should guide the choice of H, and the quantity of 'n' should be adequate to supply enough knowledge for the model to comprehend the underlying function. Naturally, there seems to be a compromise between model complexity and the size of the training data, and finding that appropriate balance is crucial for observing the "true" relationship in the data.