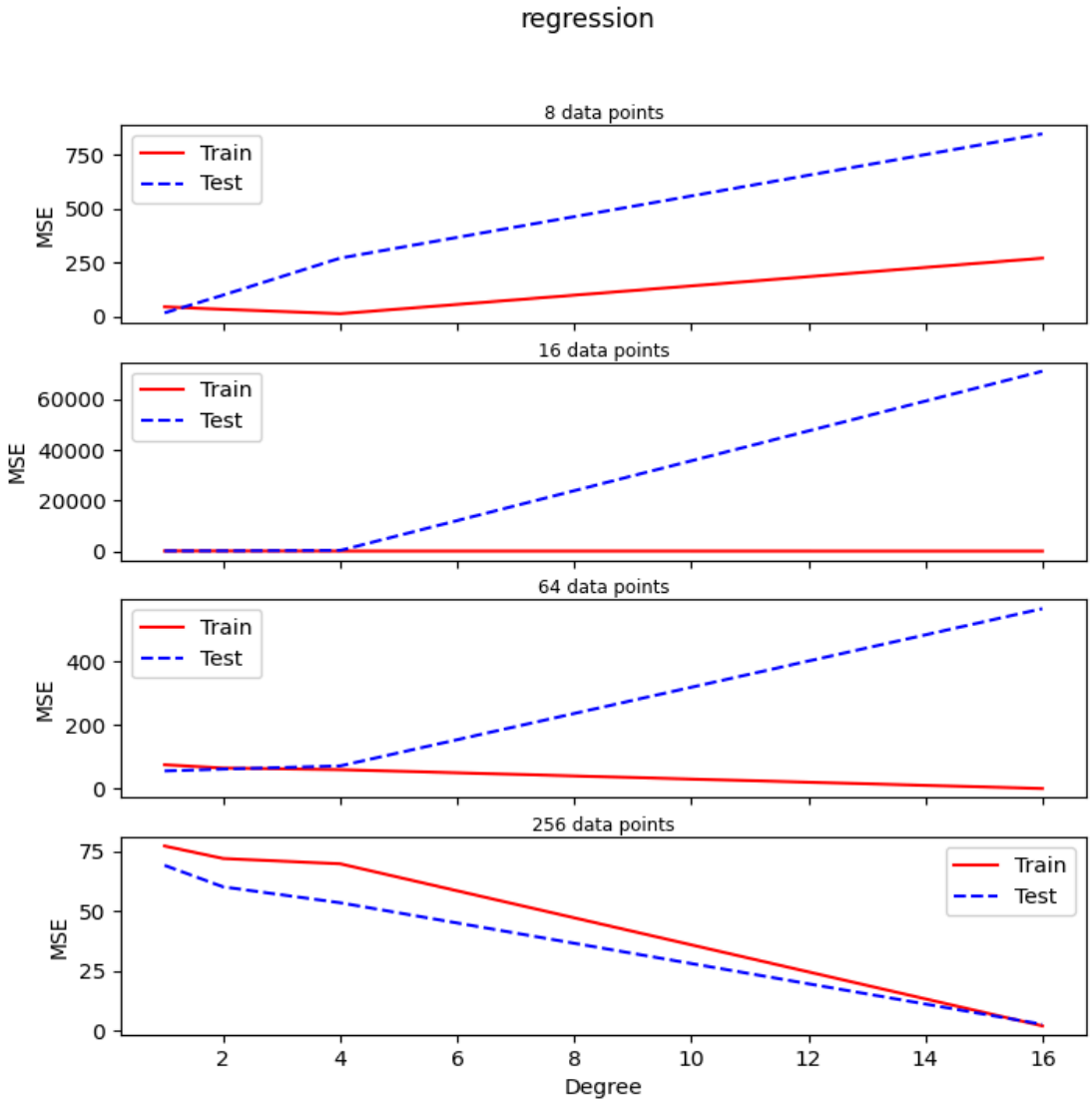
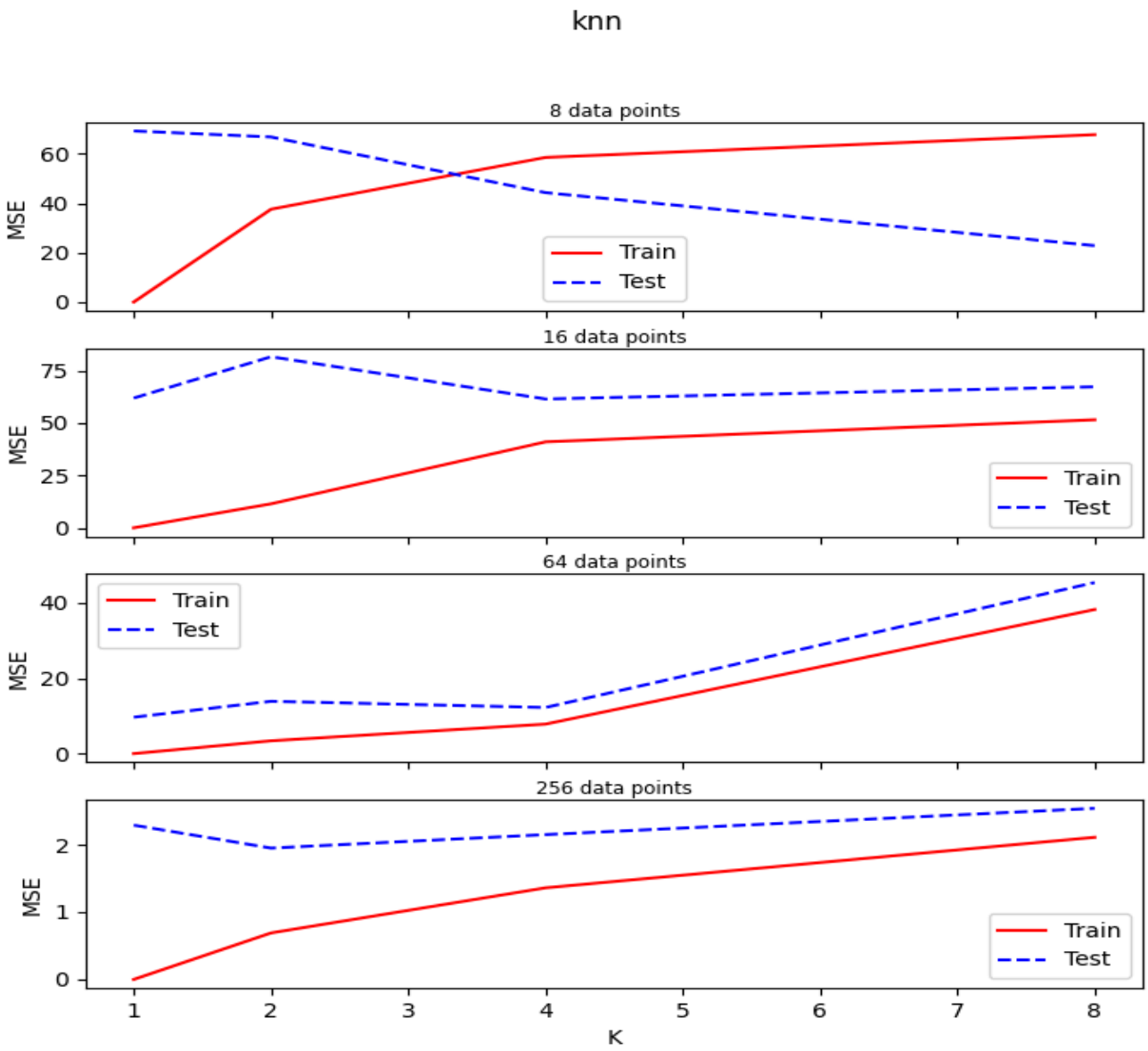
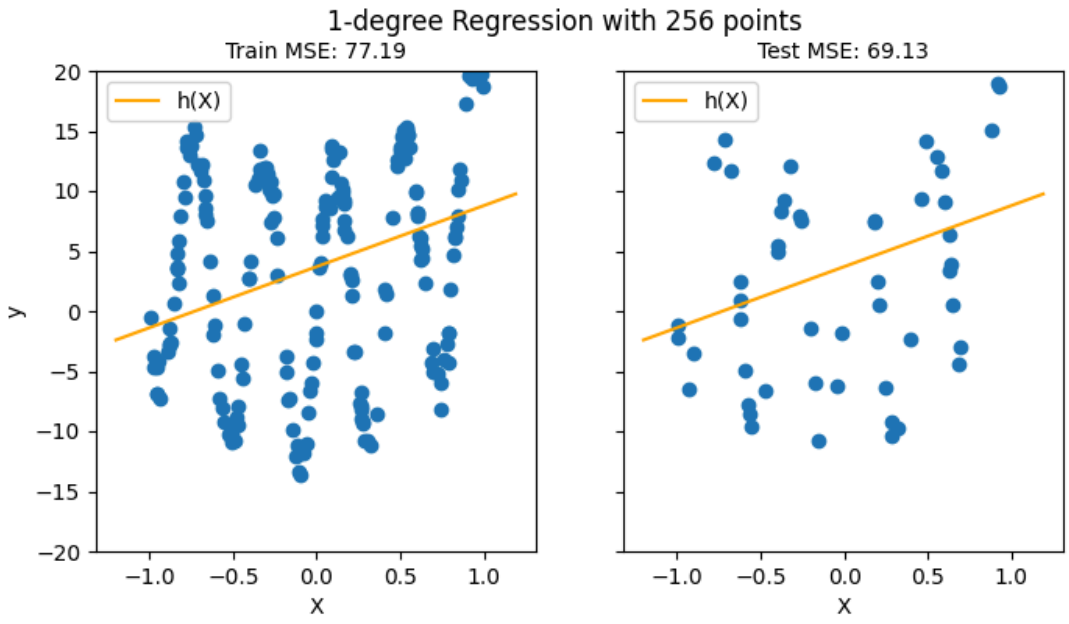


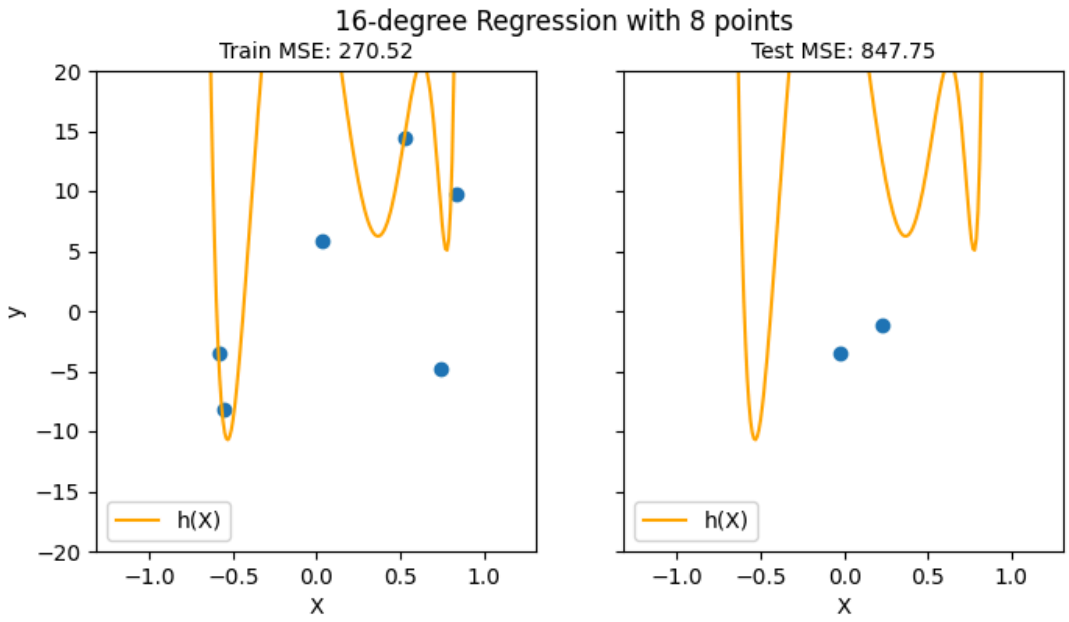
a. These plots show mean squared error varying based on the degree of the polynomial function, for a given number of data points. The Y-axis measures the mean squared error and the x-axis measures the degree of the polynomial function. For the plots with 8, 16, and 64 data points the polynomial function begins to overfit approximately for functions that have a degree that's greater than 4 degree. For the dataset of 256 data points, it begins to do a great job after around 14-16 polynomial degree function. Before that the function seems to slightly underfit for the given amount of data. The overfitting is especially observable in the plots for 16 and 64 data points. In these plots the training model perfectly passes through all the data points where the MSE for training is nearly 0. However it does a pretty bad job on the test set.



a. For KNN the y-axis measures the mean squared error and x-axis measures the amount of nearest neighbors requested in each query. There isn't a clear overfitting in the given plots, however underfitting is especially observable for the plot with 64 data points. As with the K value the generalization of the model increases, the worse the predictions of the models starts to become.



b. Here the polynomial function is clearly underfitting. This can be observed by the pattern in the data. It seems that the data has a zig zag like pattern, however the function is a linear function which can only capture linear or straight line patterns.



b. Here the function is clearly overfitting. A 16 degree polynomial function is too complicated to capture the pattern for a data set of size 8. This can be observed from where the actual data points are and how far they are from the continuous outputs or the graph of the function. Further observations can be made by imagining a mental line that captures the pattern in the data by imagining a fluctuating line that either corssos or passes nearby the data points. Then one should likely visualize how much that line will be different from the line created by the 16-degree regression line.