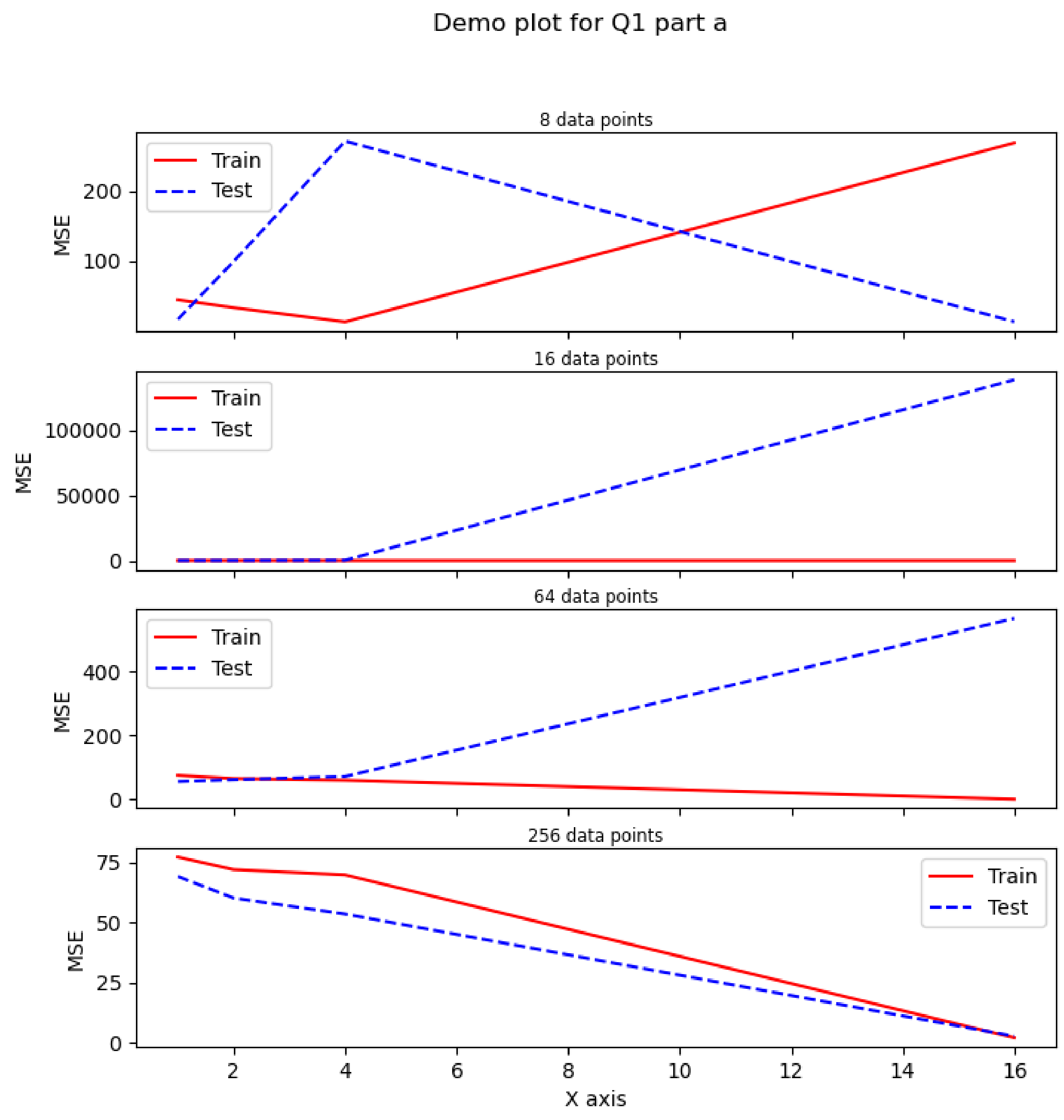
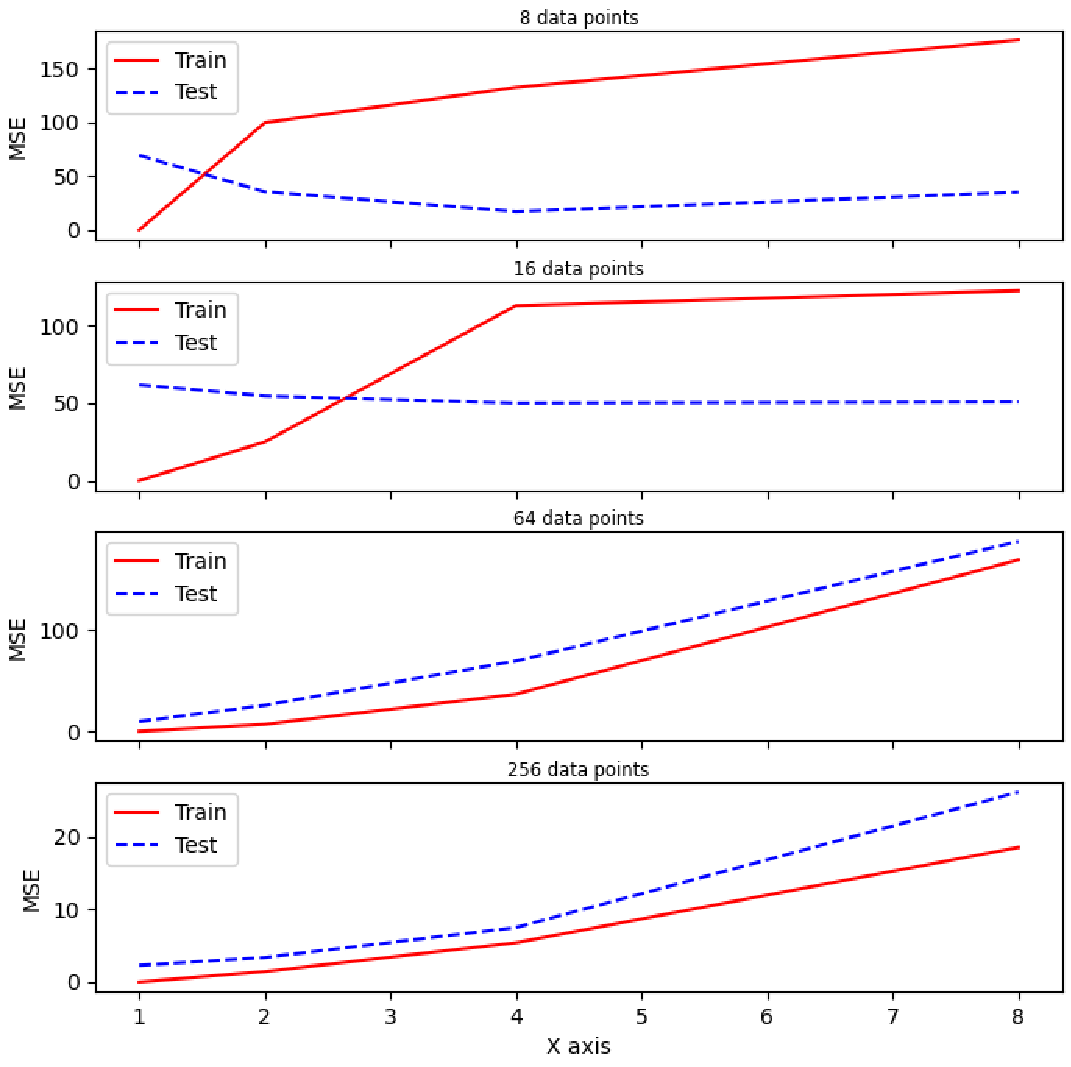
a.)



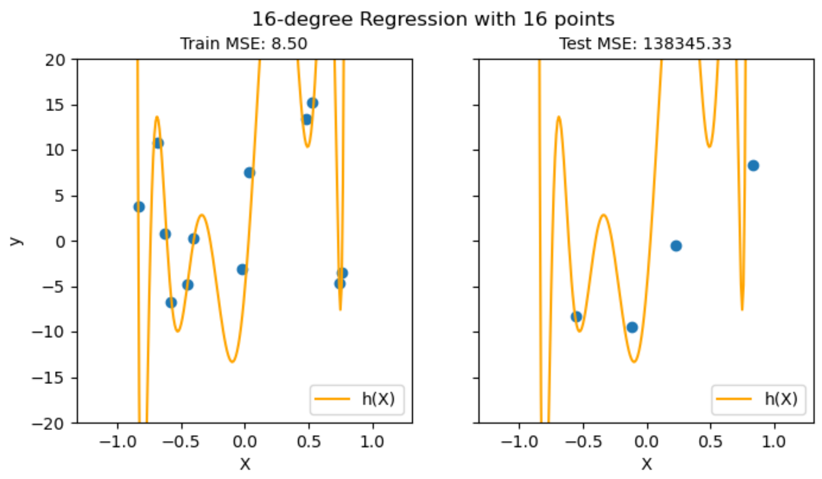
More data points means we can minimize MSE by building a better h(f), this matches what we discussed in lecture and can be seen in the graphs above. Also From the 32 graphs we can see that the data is oscillating and that when we have more data and a higher degree polynomial we can almost perfectly build a h(f) that minimizes the MSE for the train data and the test data.

Demo plot for Q1 part a



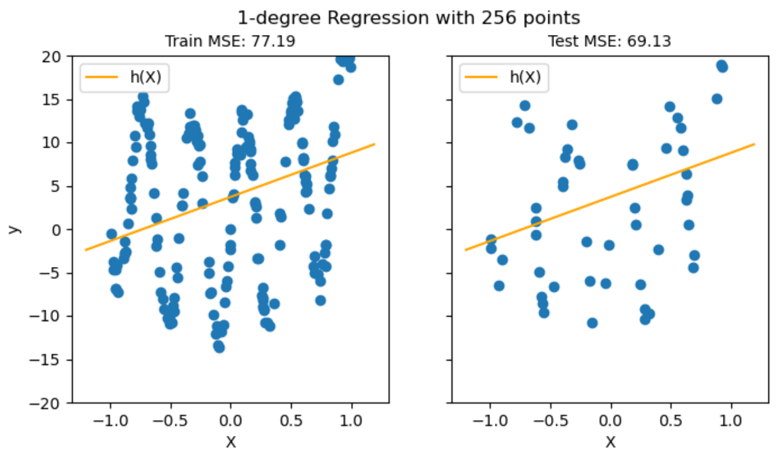
From the graphs above we can see that when the amount of data increases then MSE decreases, because we can train a h(f) that better approximates f(x) We can also see that when n_neighbors is increasing our MSE is also increasing, because the model is undefitting the data, but when n_neighbors is small then we can train a h(f) that very closely approximates the f(x) and minimizes the MSE which corresponds with what was discussed in lecture.

b) **Overfitting**:



In the train data plot we can see that the data is ‘perfectly’ fitted with the hypothesis function, but this will not generalize well to test data points, because the hypothesis function that was learnt is too specific.

Underfitting:



In the train data plot we can see that the train data oscillates, but the function that we are fitting to the data is a straight line and thus we cannot get an accurate hypothesis function. Thus test data will have incorrect predictions.

c) When f(x) is a high degree polynomial and we choose our H class to be some lower order polynomial then we will not be able to generate a h(x) that approximates the f(x) and this will cause underfitting as the polynomial degree of h(x) is lower than the f(x).

When f(x) is a low degree polynomial and we choose our H class to be some higher order polynomial then we will not be able to generate a general h(x) that approximates the f(x). Our h(x) will be too specific and will not perform well on test data. This will cause overfitting as the polynomial degree of h(x) is higher than the f(x).

When n training examples increase we can better generate a h(x) that approximates the f(x)