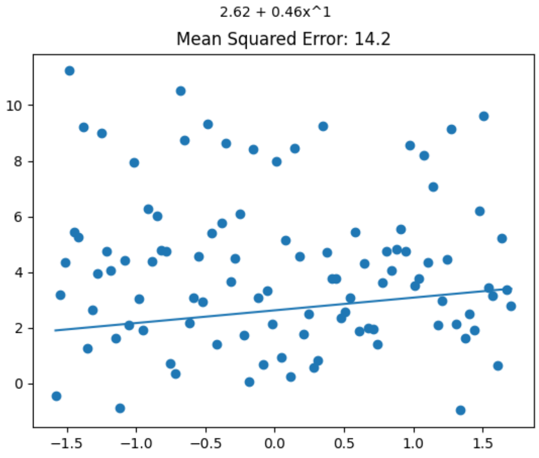
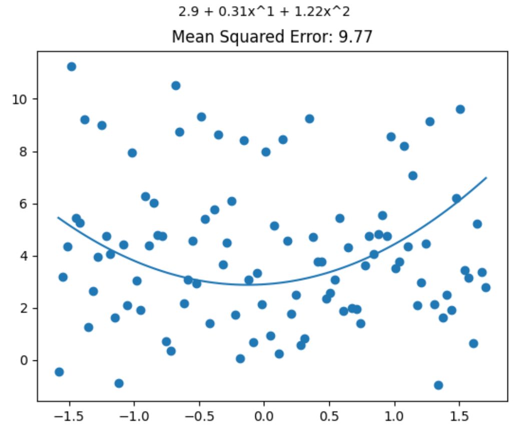
For this project I chose to use full batch gradient descent in my algorithm.

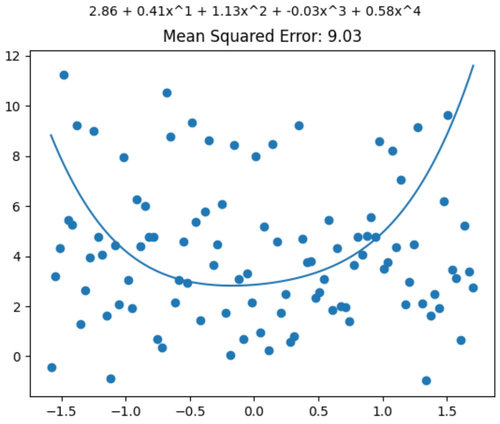
My alpha value is 0.001, and I use 1000 iterations during training.

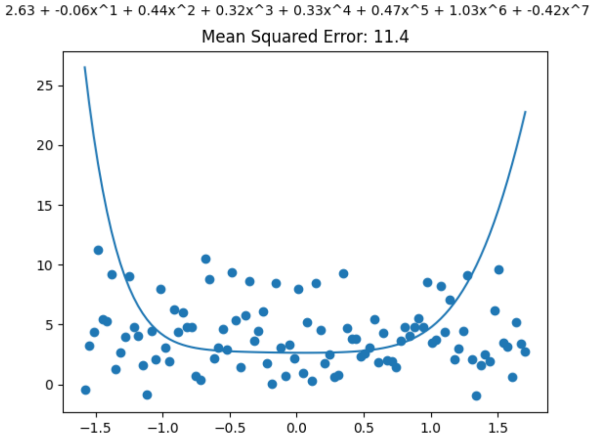
My Project makes much use of the numpy module. Numpy does most of my complex calculations through the manipulation of matrices so that the program runs more efficiently. See final results below.

Synthetic-1:



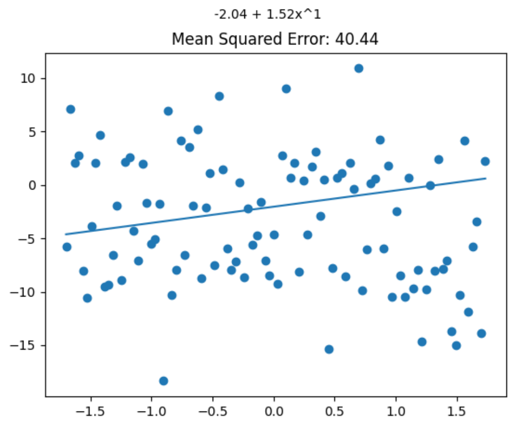


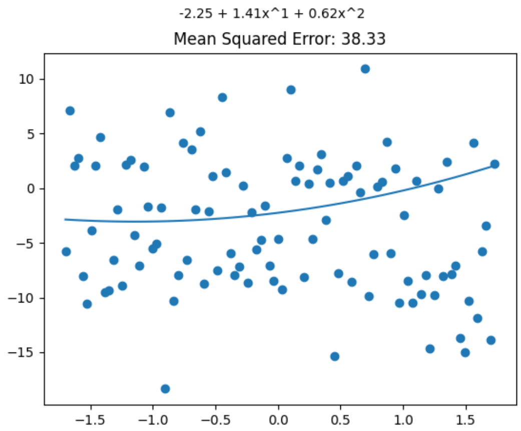


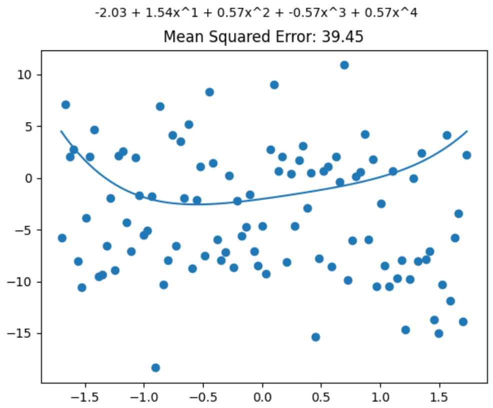


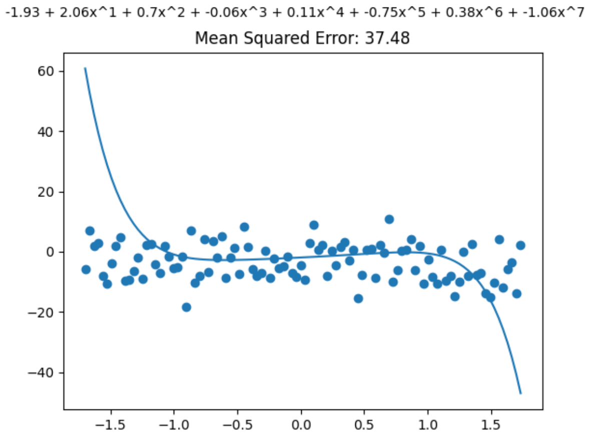
Interestingly it seems the 7th order polynomial did not reduce error, but rather increased it to the point where it was almost as inaccurate as the 1st order.

Synthetic-2:



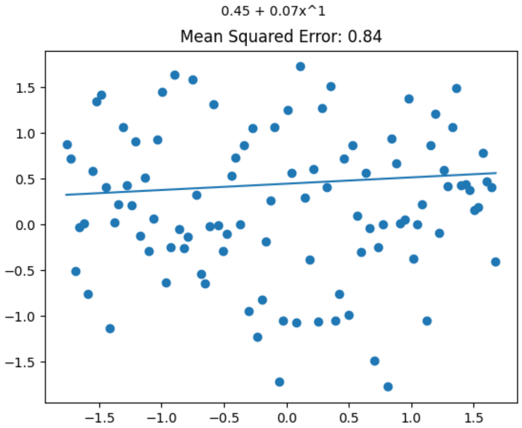


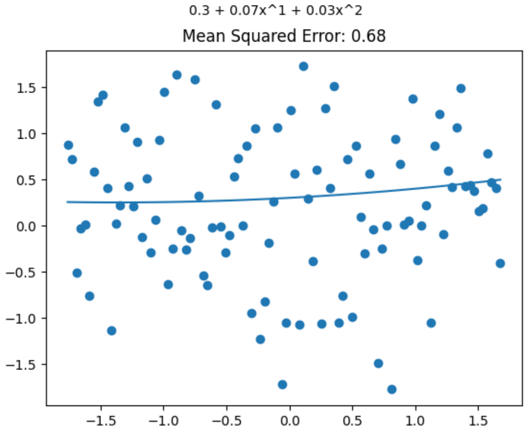


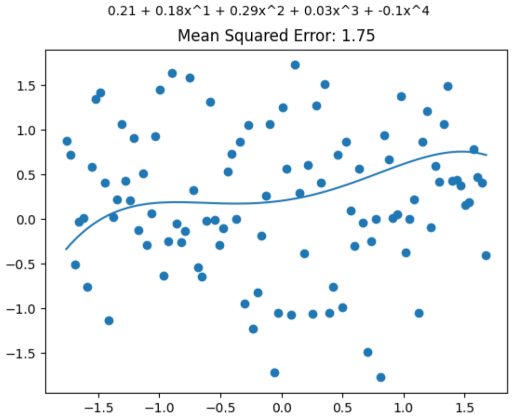


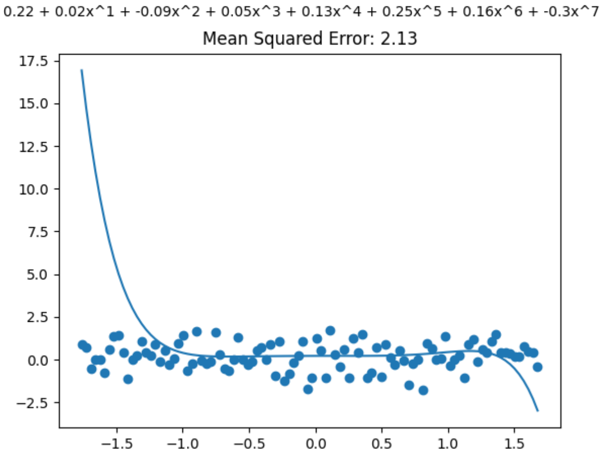
Once again, I am surprised to see that a higher order function does not always help increase accuracy. This time around the 7th order polynomial seemed to be the best fit.

Synthetic-3:









Unfortunately for some reason it seems my program does not like synthetic-3, as I only met benchmark on one of the graphs. However, I was well within benchmark on the other files, so as to why it would be more inaccurate on one specific file I cannot say.

References

https://rickwierenga.com/blog/ml-fundamentals/polynomial-regression.html