**Coding\_1**

1. python3 Regression.py --data linear --validation --display

Best Polynomial: 1

Training mean square error: 0.05340004614270319

Test mean square error: 0.028905433278947075

1. python3 Regression.py --data quadratic --validation --display

Best Polynomial: 2

Training mean square error: 0.0075842419667236675

Test mean square error: 0.009004923866326916

1. python Regression.py --data unknown --validation --display

Best Polynomial: 5

Training mean square error: 5.2313718769164274e-06

Test mean square error: 9.55551289922081e-06

1. python Regression.py --data unknown\_noise --validation --display

Best Polynomial: 1

Training mean square error: 0.15841514051470137

Test mean square error: 0.15742128951764817

Discussion:

For the unknown\_noise file, the difference between the error of training and testing data is big from poly1 to poly12. That is because the noise indeed intervened the prediction. Therefore, the best poly is 1 which has the smallest loss.

For the unknown file, every data seems close to the regression curve which means there is little loss for poly 5, compared to other polys. Therefore, the best poly is 5.

**Coding\_2**

Best dis\_metric: cosine

Best K: 1

Best Leave-one-out accuracy: 0.9979661016949153

Test accuracy: 1.0