Assignment 1 CS4442

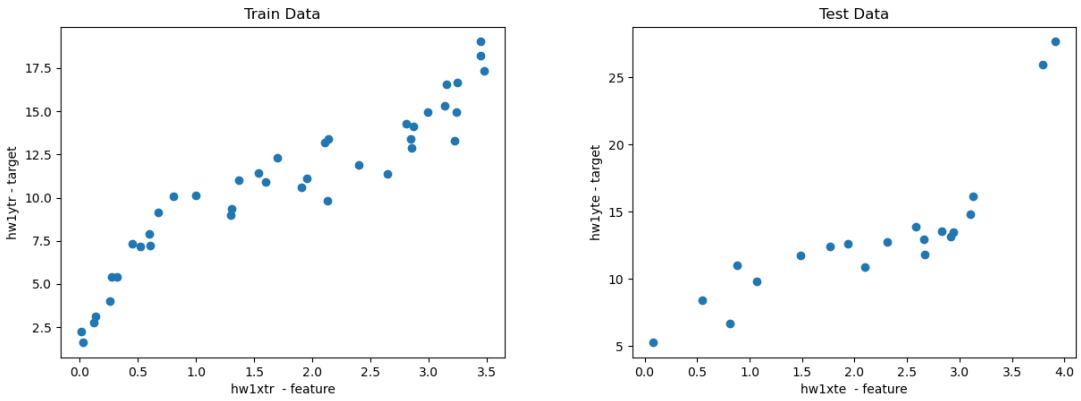
Nicholas Pysklywec

March 6, 2023

1 Linear and Polynomial Regression

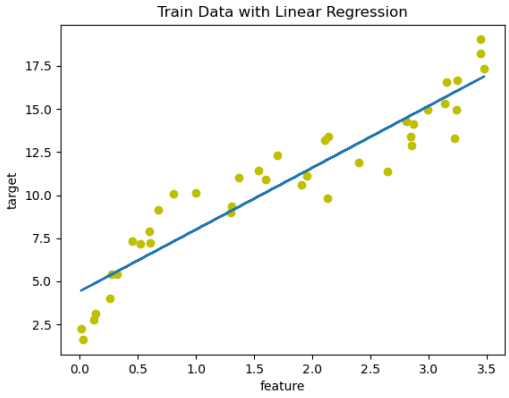
Part A

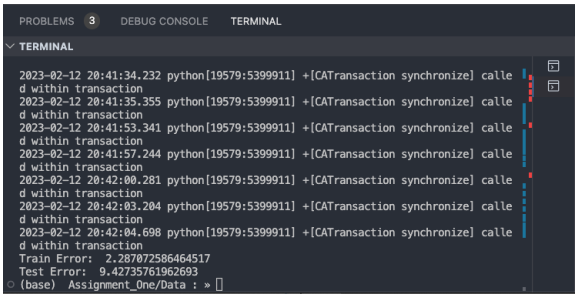
The python file question2a.py includes the specific code for this answer. Plotting the train and test data respectively yields the following:



Part B

The python file question2bandc.py includes the specific code for this answer. The linear regression line plotted with the training data, alongside the average training error:

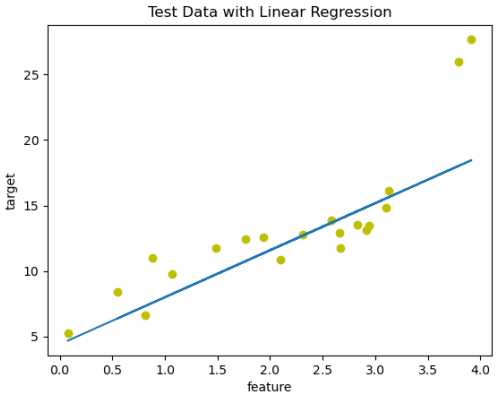


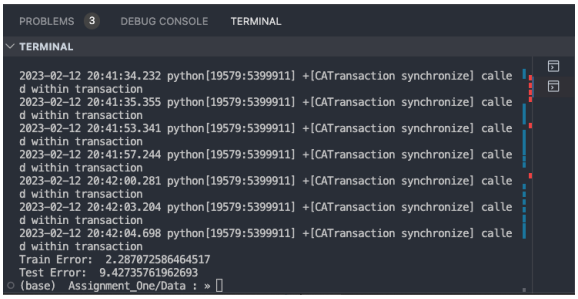


The average train error is 2.29.

Part C

The python file question2bandc.py includes the specific code for this answer. The linear regression line is plotted with the test data, alongside the average testing error:

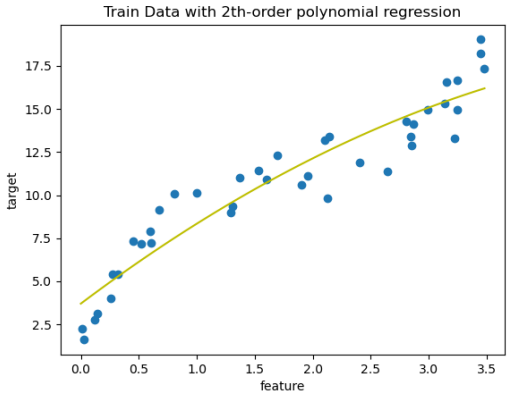


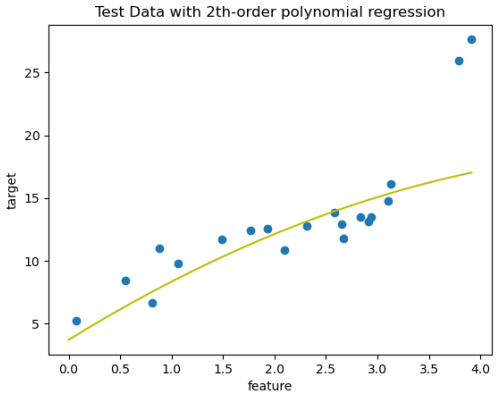


The average test error is 9.43.

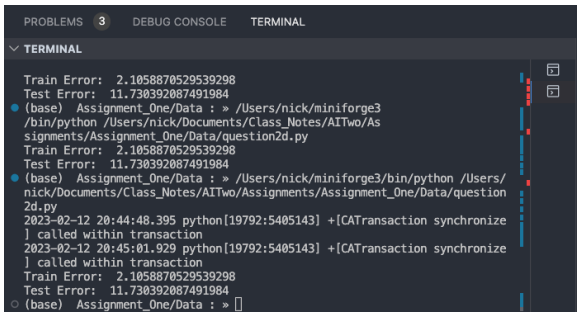
Part D

The python file question2d.py includes the specific code for this answer. The model is overlayed on two plots, train, and test respectively.





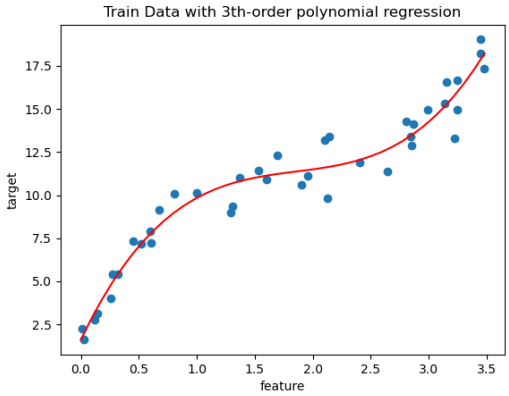
Additionally we receive train and test error, seen below:

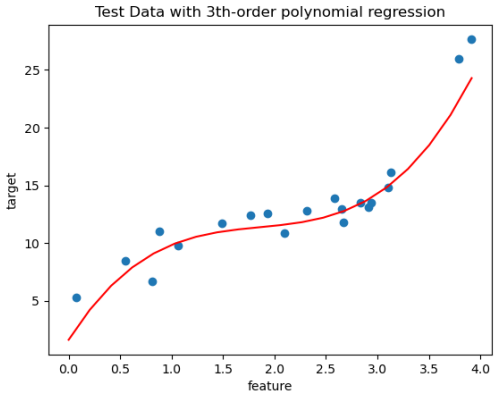


The train error is 2.10, the test error is 11.73. The training error is less, however this comes at a cost of a much increased test error. This signifies this model is not a good fit for the data. 2nd order polynomial regression did not improve performance.

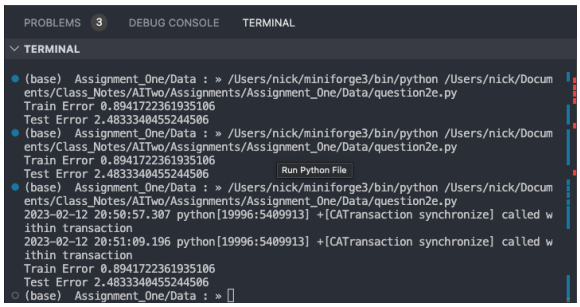
Part E

The python file question2e.py includes the specific code for this answer. The model is overlayed on two plots, train, and test respectively.





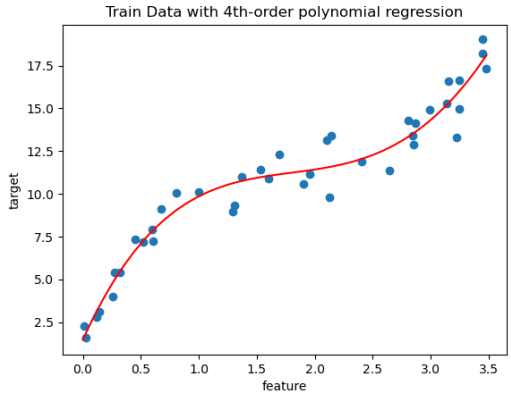
Additionally we receive train and test error, seen below:

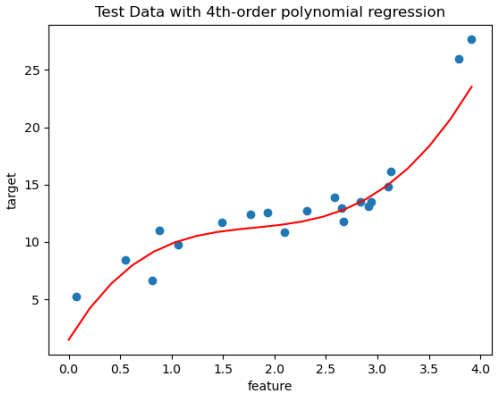


The train error is 0.89, the test error is 2.48. These values are good, the train error is significantly lower than the two previous models, and the test error more importantly is a much lower value. We can say the 3rd order polynomial regression model suits the data the best so far, as it reduces the average error the most.

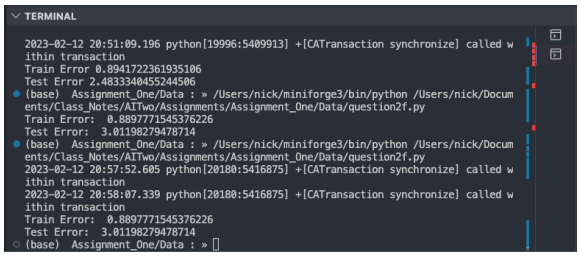
Part F

The python file question2f.py includes the specific code for this answer. The model is overlayed on two plots, train, and test respectively.





Additionally we receive train and test error, seen below:

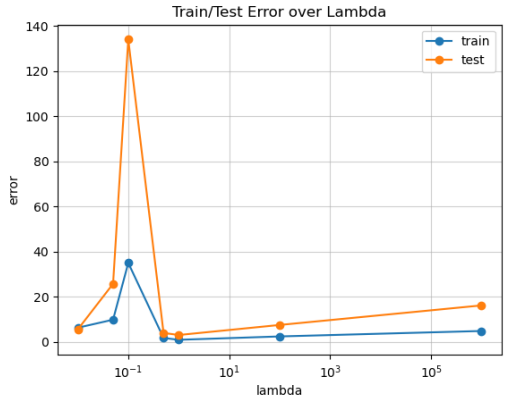


The train error is 0.89, the test error is 3.01. The test error is worse then the error received for the 3rd order polynomial regression model. Therefore, the 3rd order polynomial regression model fits the data the best.

2 Regularization and Cross-Validation

Part A

See code for this question in question3.py. The different lambda values are iteratively used with the 4th order polynomial regression model. The train/test values associated with the different lambda values used were:



The lambda value that is the most ideal is 10. The train and test error is certainly the lowest on the plot. 10 is the best lambda for fitting the 4th order polynomial regression model to the test and train data.

Part B

Each of the weight values are plotted below as they change with each new regularization parameter lambda: