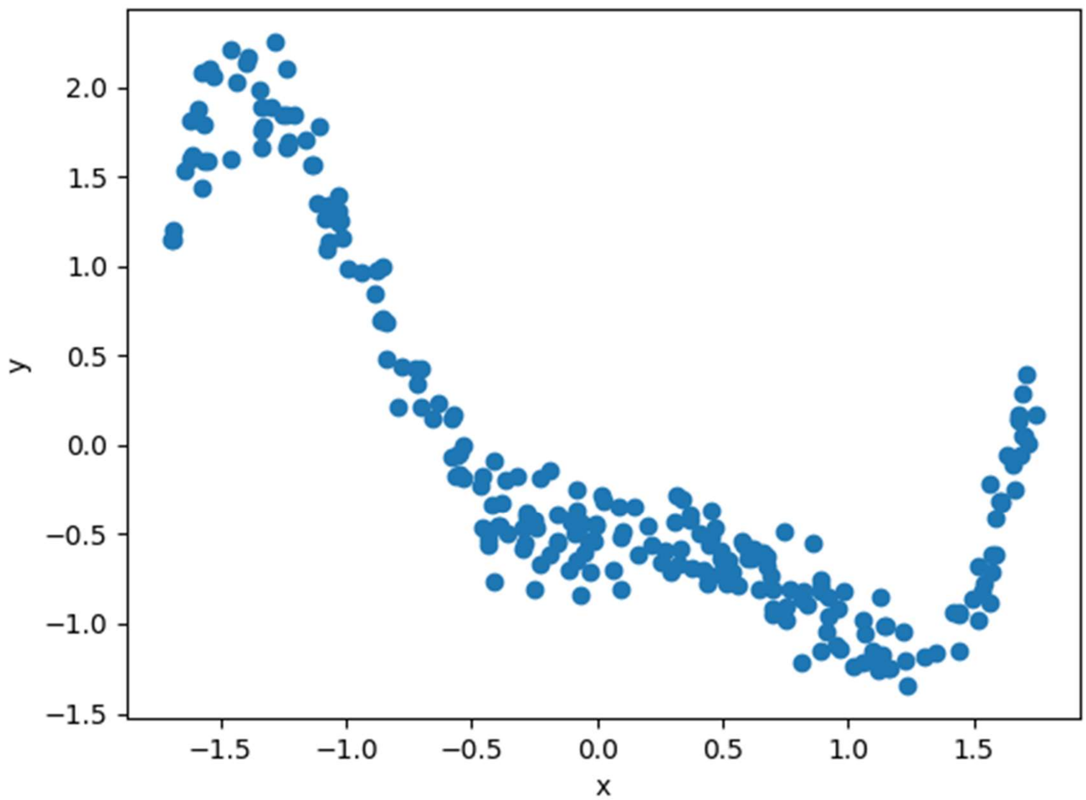
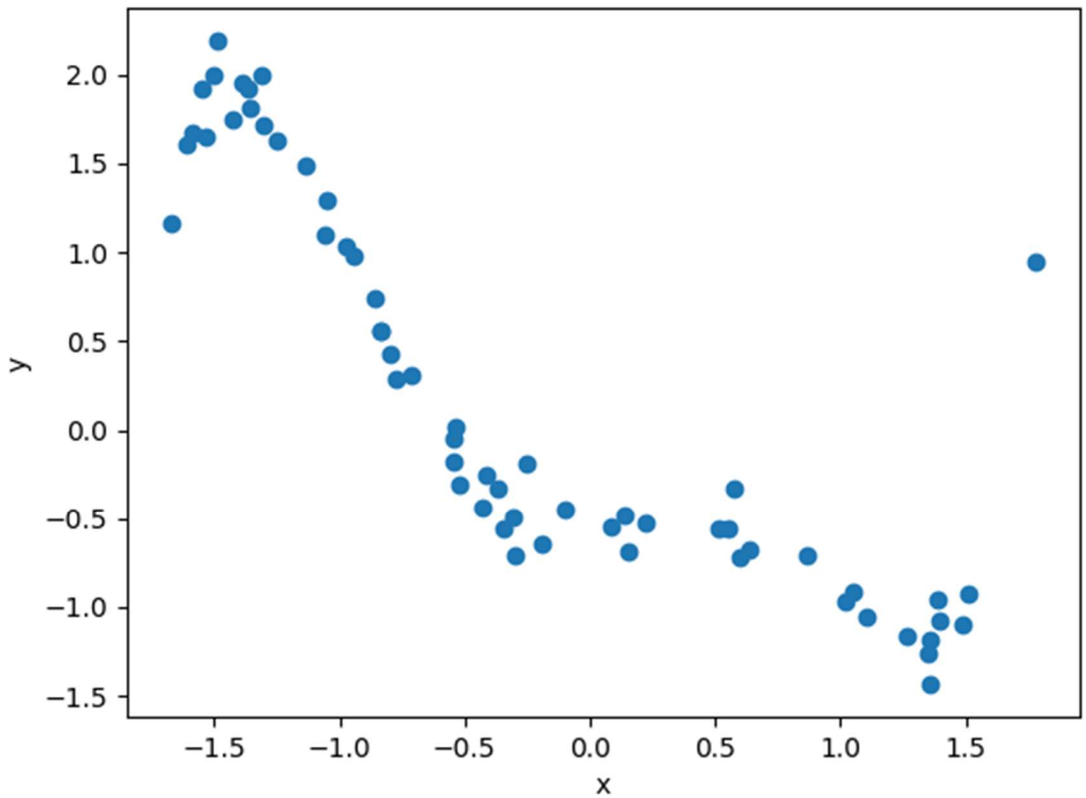
Polynomial Regression

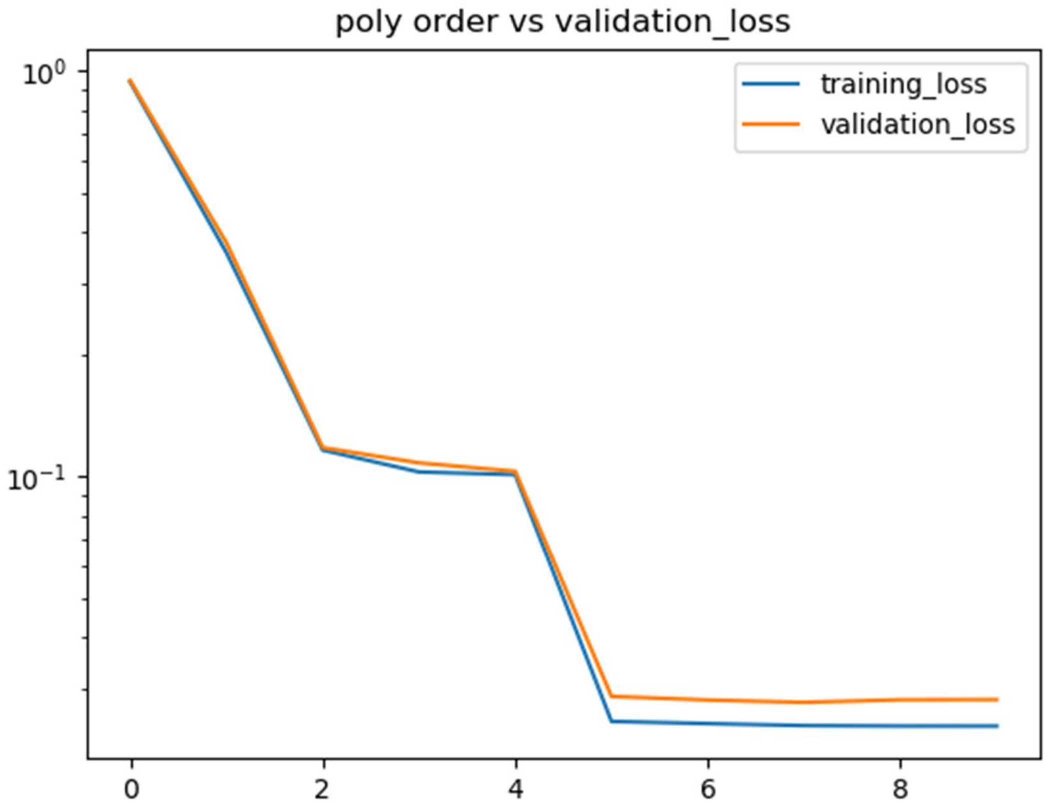
```
Hyperparameter Selection
Best Degree: 7
Best Theta: [-0.53103609 -0.02168913  0.57052874 -1.71309263  0.0370365   0.6737321
 -0.03256637 -0.03962692]
Best Theta:  [-0.53103609 -0.02168913  0.57052874 -1.71309263  0.0370365   0.6737321
 -0.03256637 -0.03962692]
Test Loss: 0.02125312187756854



Gradient Descent
Best Degree: 7
Best Theta: [-0.35529357 -0.4165501   0.15522872 -0.41374686  0.20231069 -0.29417689
 -0.04526306  0.15937102]
```



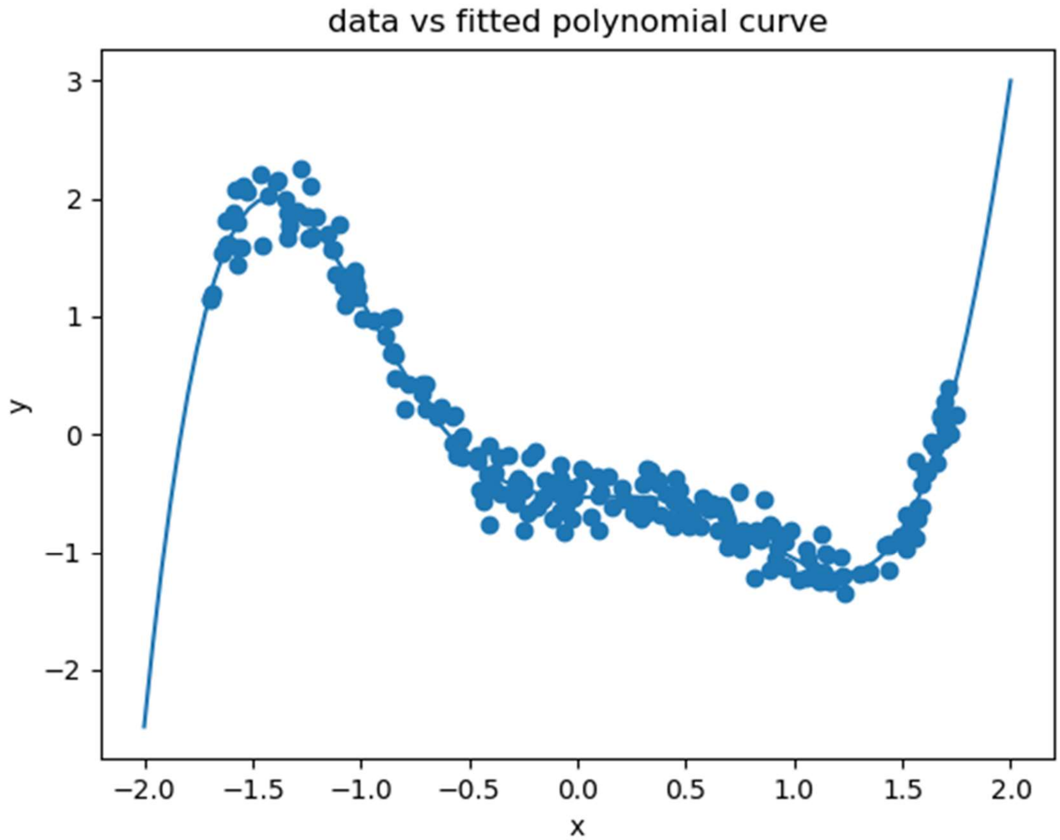
A plot of the training portion of the generated dataset



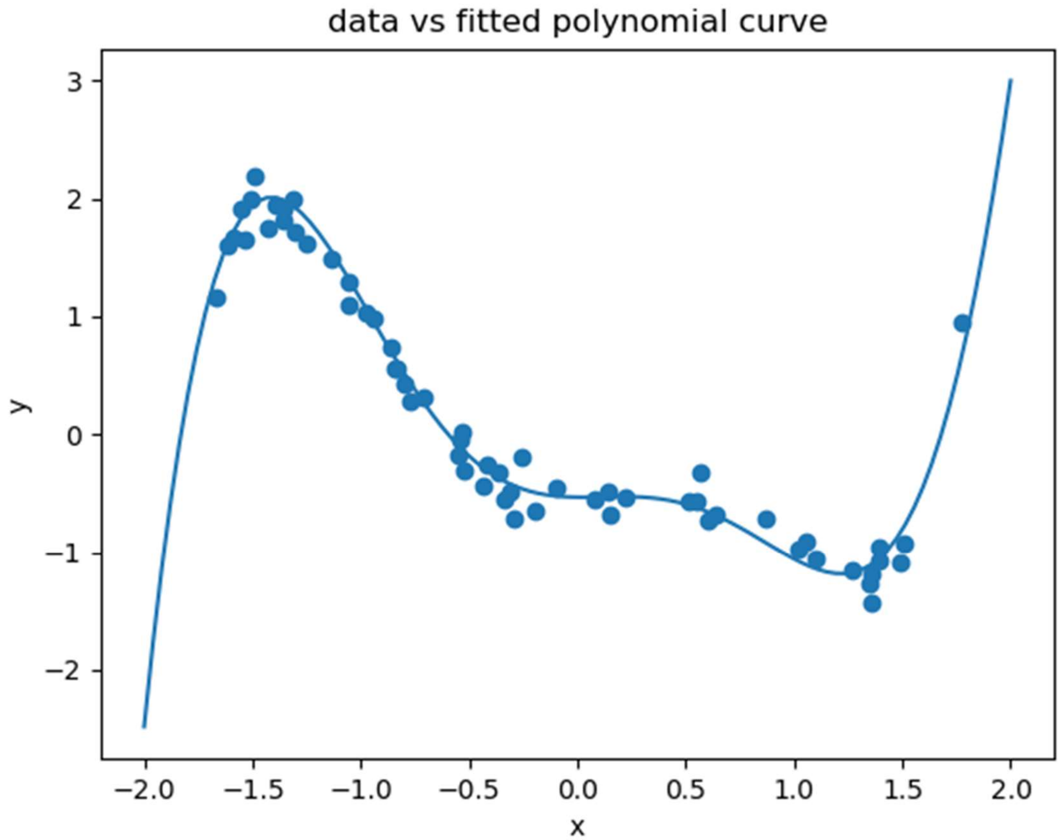
A plot of the test portion of the generated dataset



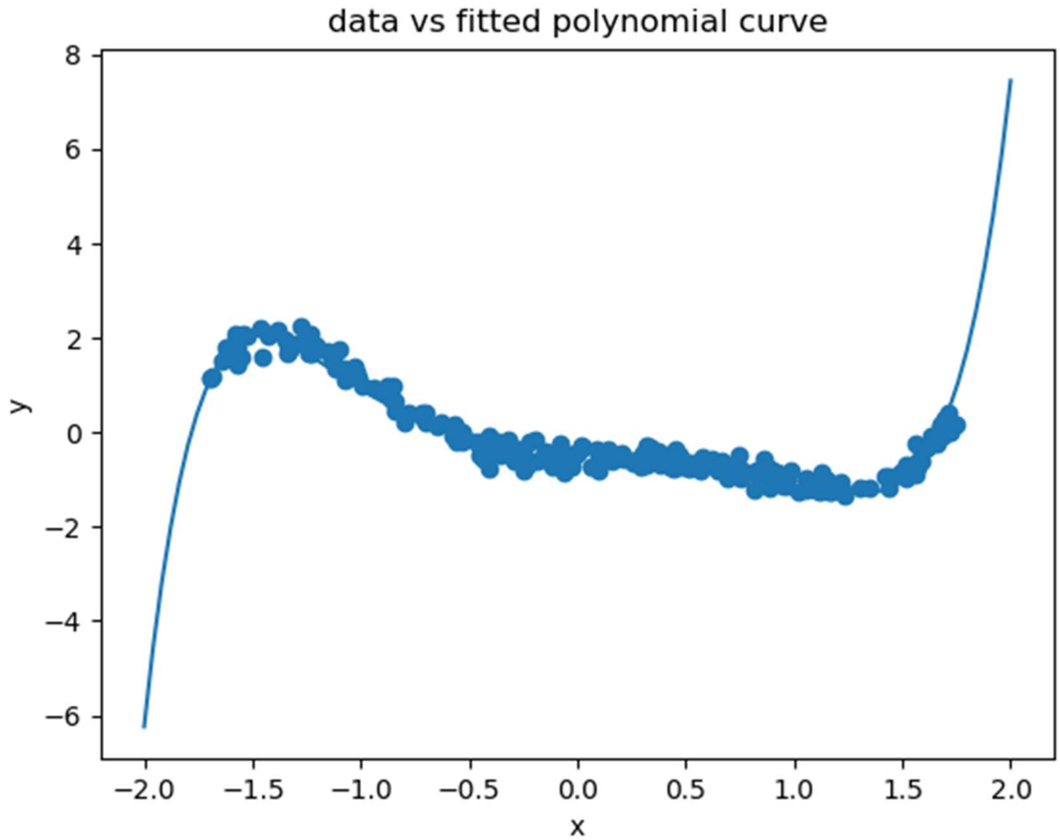
A plot of the order of the polynomial transformation compared to the training and validation losses; Degree 7 is found to be the best degree (since it has the minimum loss in validation). This is a bit odd considering the data was generated using a fifth-degree polynomial and would suggest that slight overfitting has occurred.



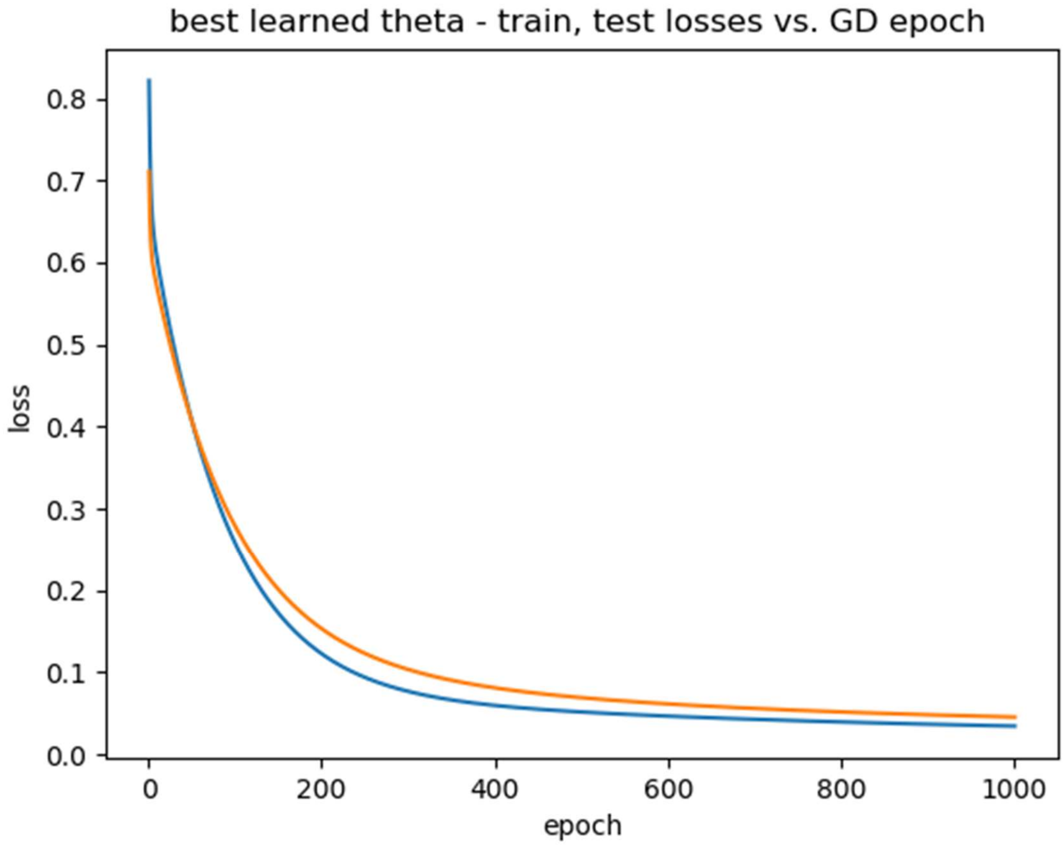
A plot of the best-fit polynomial curve over the training data



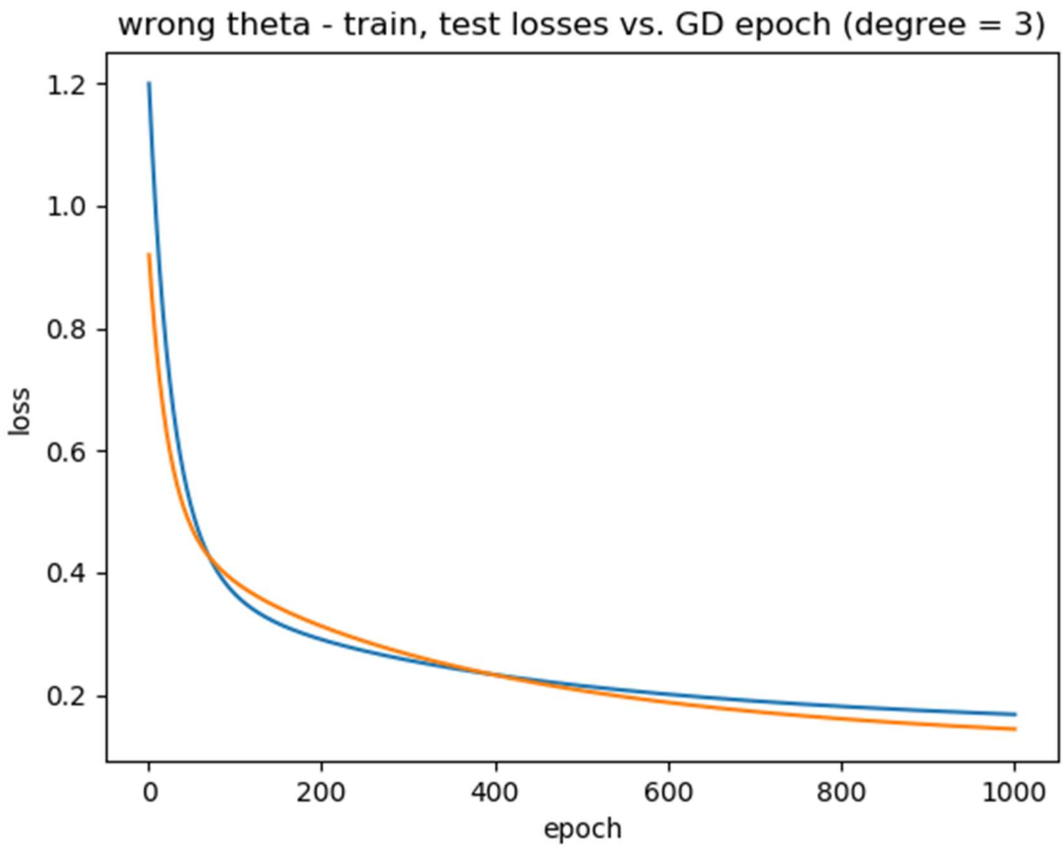
A plot of the best fit curve over the test data. The best-fit theta was derived using the normal equation with a degree of 7, since that was found to be the best degree during hyper-parameter tuning.



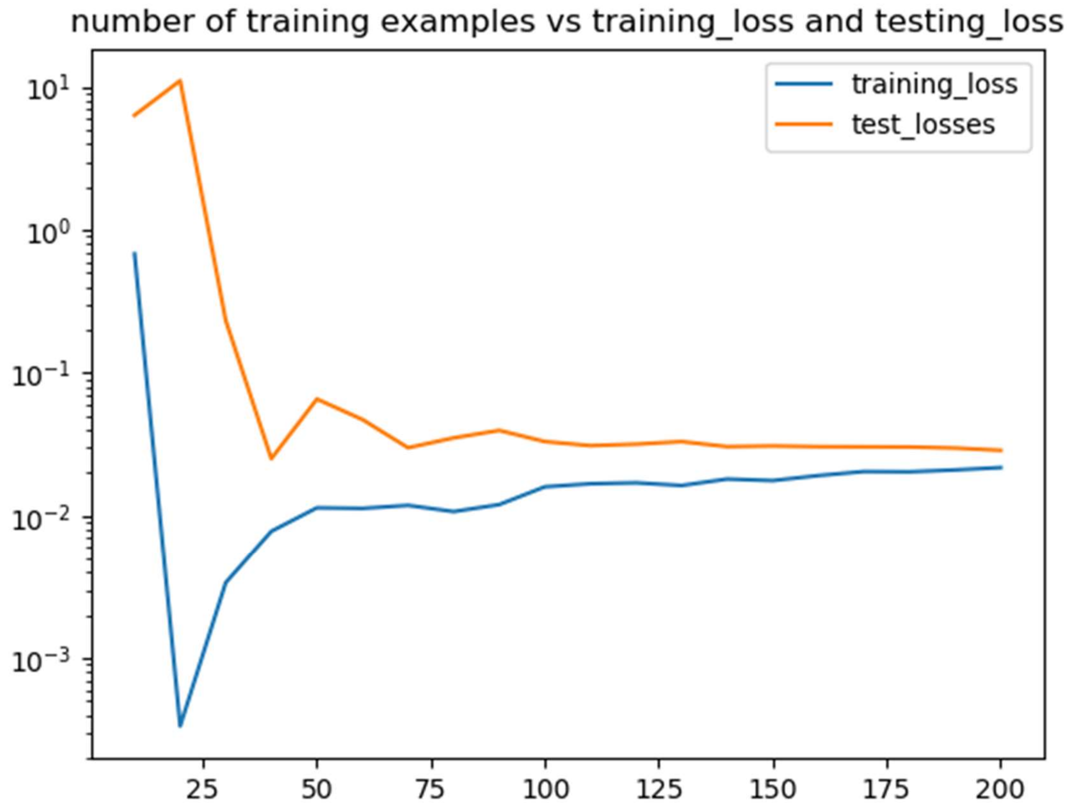
A plot of the curve found via gradient descent over the training data with optimal degree and theta. *I'm not really sure what is meant by "we require you to generate a similar plot for one more hyperparameter degree d from part 2, that is not the best hyperparameter." This is pretty ambiguous.*



A plot of the gradient descent loss during the training with the optimal degree (degree = 7)

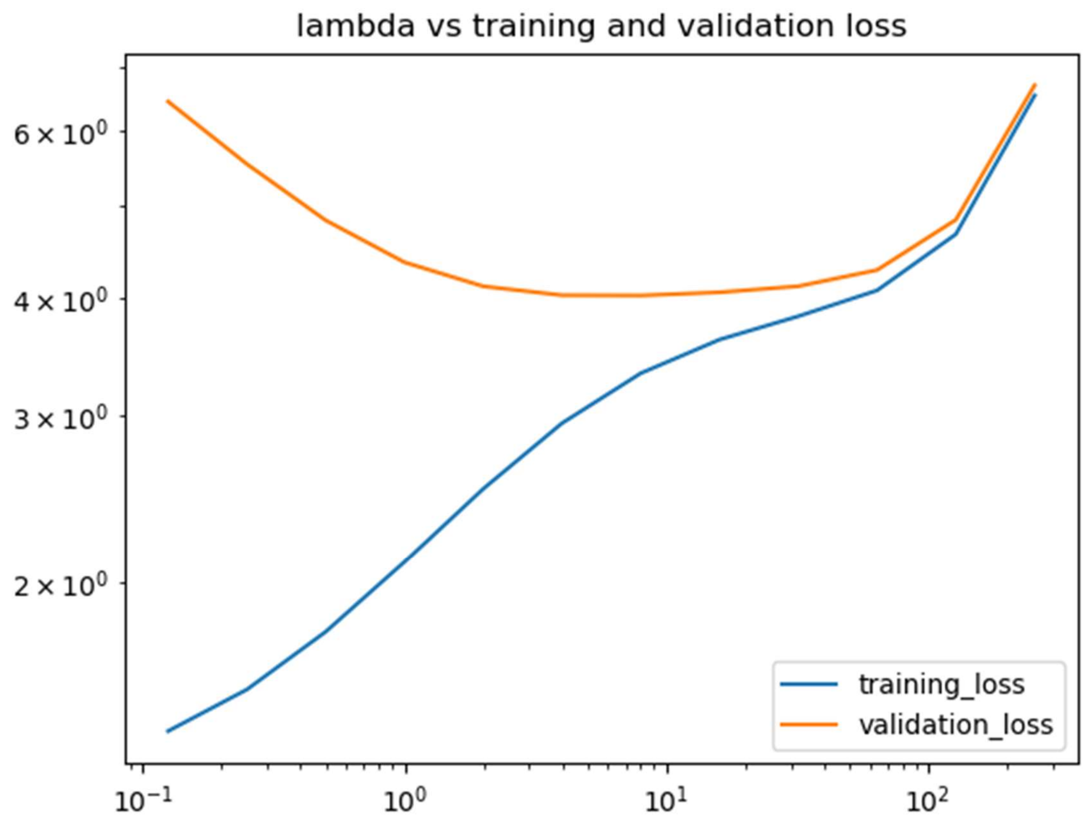


A plot of the gradient descent loss when the wrong degree is chosen (degree = 3). As you can see, the error to which the algorithm converges is an order of magnitude higher than the previous graph.

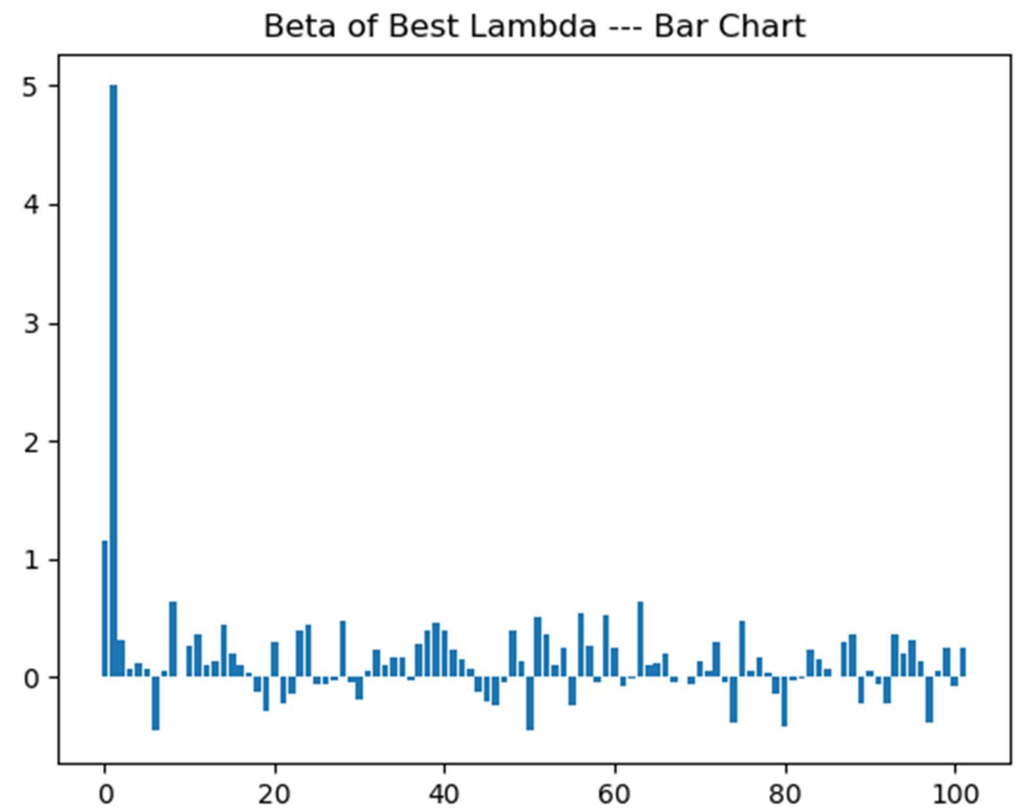

number of training examples vs training_loss and testing_loss
training_loss
test_losses
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
25
50
75
100
125
150
175
200

Ridge Regression

```
Best Lambda:  8
L2 norm of normal beta:  30.2696547988007
L2 norm of best beta:  5.748383109215213
L2 norm of large lambda beta:  5.748383109215213
Average testing loss for normal beta:  11.031286619642902
Average testing loss for best beta:  4.655100367769717
Average testing loss for large lambda beta:  12.12642033283745
```



The plot shows that there is clearly a “golden zone” for lambda that produces the lowest average validation loss around 8.



A plot of the beta values. This makes a lot of sense given that the underlying distribution is 5x+3, with a bunch of random noise tacked on for the latter 100 or so variables. You can clearly see that the model "figured out" the underlying distribution (for the most part).