Assignment 2: More Regressions and Model Selection

UVA CS4774

March 6th, 2022

Owen Richards

Polynomial Regression Model Fitting:

Task 1 hyperparameter tuning:

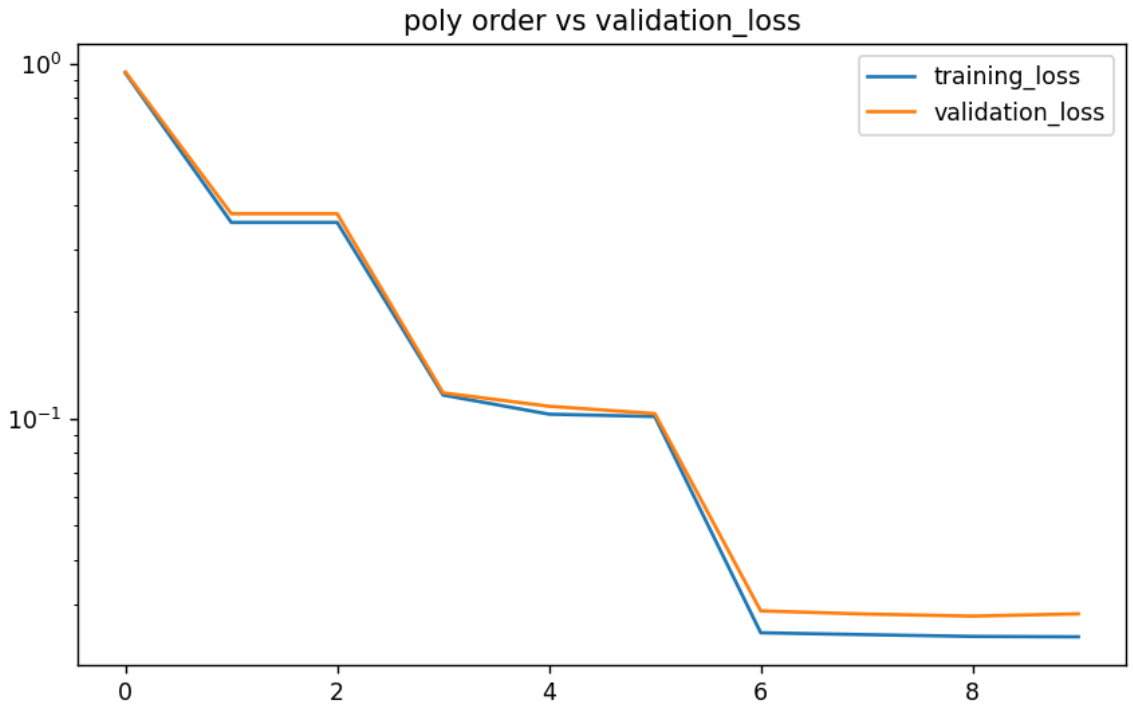

Fig 1: Polynomial Order versus Training and Validation Loss

For the polynomial regression model fitting, 300 samples were used for this task. In Fig 1, 60% of the total data was used as training data, 20% of the total data was used for validation data and the remaining 20% was for testing. The train data is used to create a model fit, whereas the validation set is used for model selection and hyperparameter tuning. The validation set can estimate the future performance of the model by performing multiple regressions and choosing the one with the smallest error. One of the negatives of using a validation set is that it does mean that some of the data that would have been used for training is now used for validation. Also, if it is a small amount of data to begin with it might be the case that the validation-set might just be lucky or unlucky and not a true estimate of future performance. In the figure above, the x axis is

the value of d, and the y-axis is the MSE loss on the training set.and validation set. The best hyperparameter was a d = 8. This means the polynomial regression with the hyperparameter being 8 has the best expected performance.

Task 2:

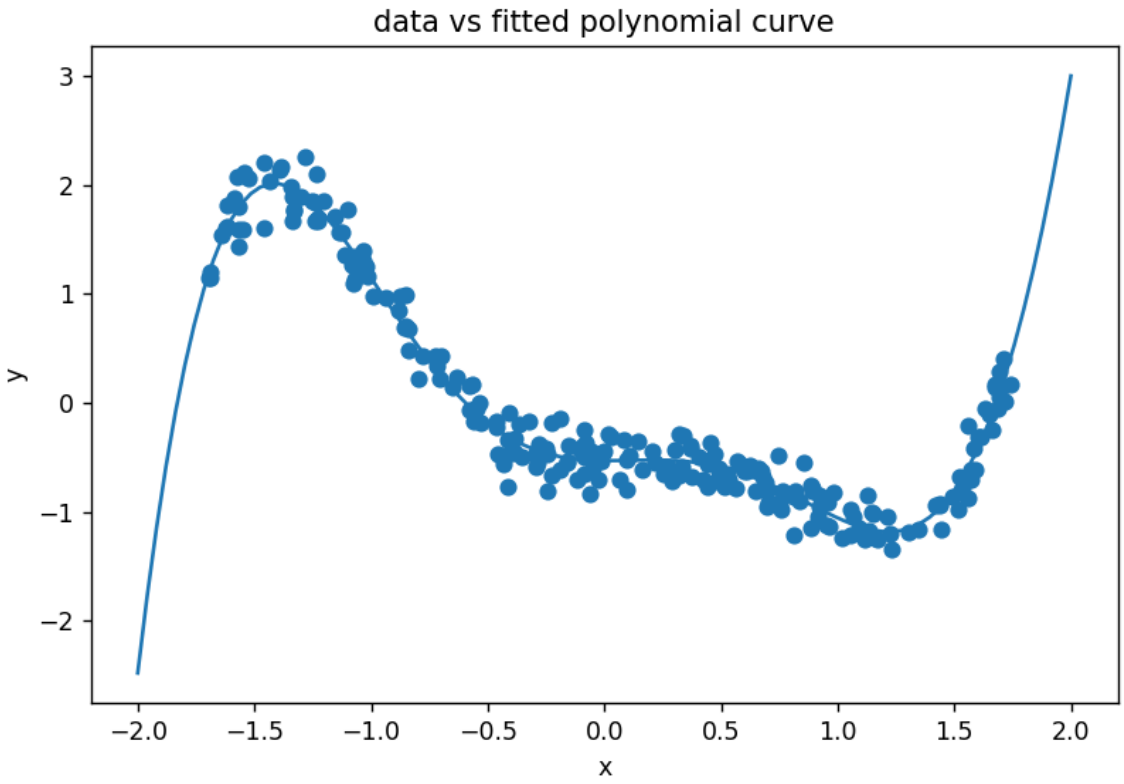

Fig 2: Training Data versus Fitted Polynomial Curve (d=8)

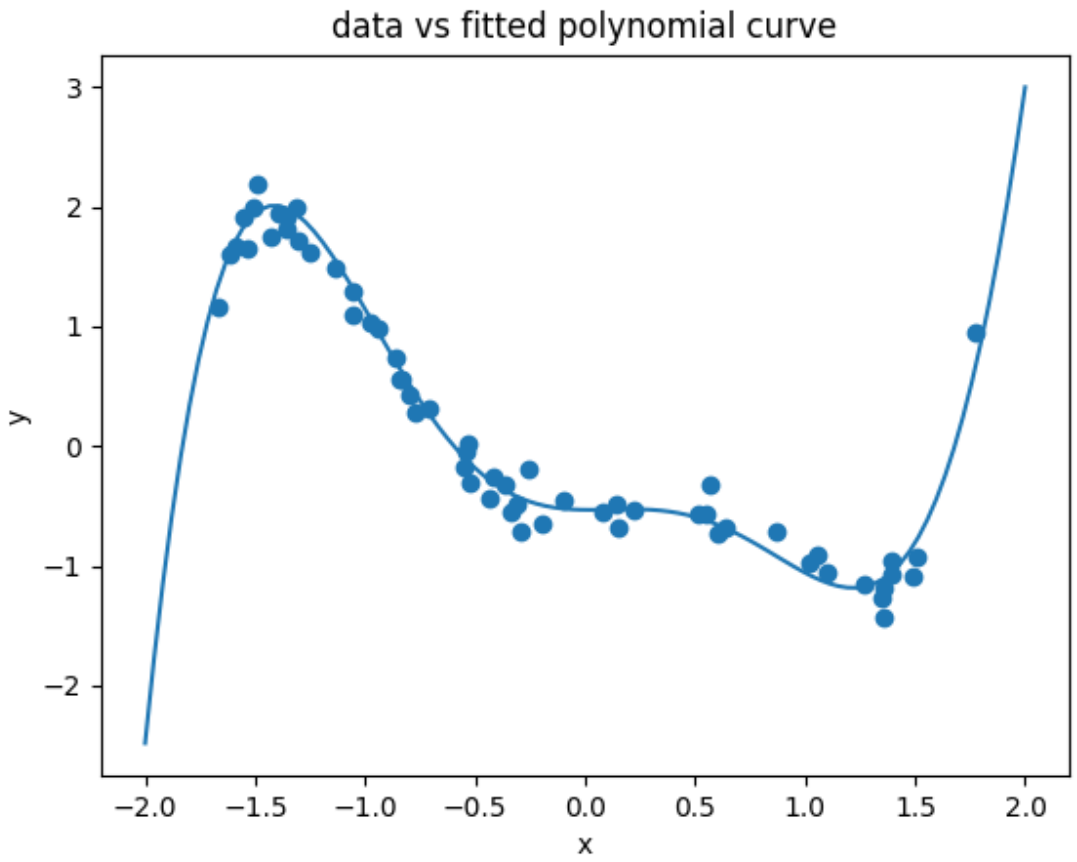

Fig 3: Test Data versus Fitted Polynomial Curve (d=8)

The best fit curve equation is: $-0.03962692x^7 + -0.03256637x^6 + 0.6737321x^5 + 0.0370365x^4 + -1.71309263x^3 + 0.57052874x^2 + -0.02168913x^1 - 0.53103609$. The final test MSE is 0.021253121877552655. Finally, the best theta is [-0.53103609 -0.02168913 0.57052874 -1.71309263 0.0370365 0.6737321 -0.03256637 -0.03962692]. The best theta was found with the best polynomial basis based reformulation of the x training set and the y training set. The reformulating x training set and the y training set were then run in the normal equation to produce the best theta. The reformulated x training set was done by using the best degree which was 8. As shown above in Fig 2 and Fig 3, the learnt theta makes sense in relation to the true underlying distribution since it is such a good fit to both the training and testing data.

Task 3:

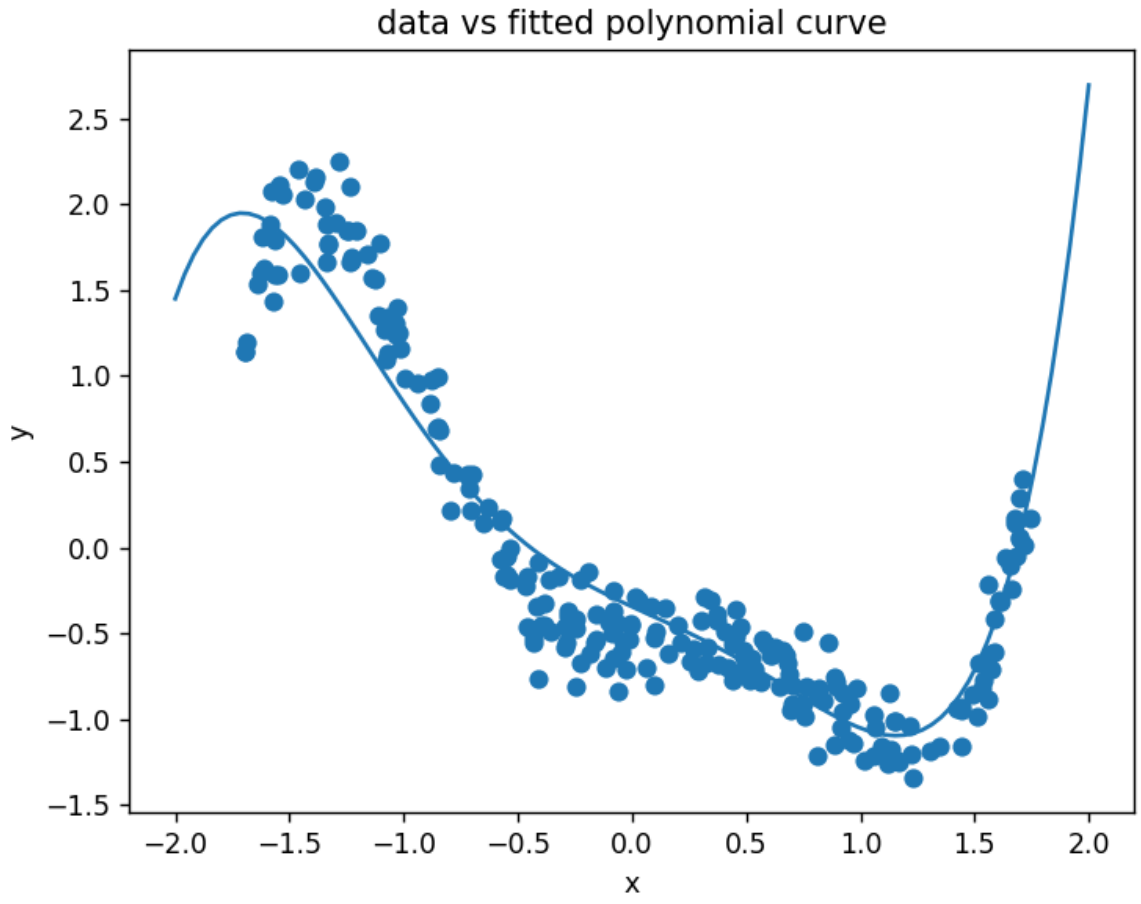

Fig 4: Training Data versus Fitted Polynomial Curve using Gradient Descent Optimization

($d=6$, $t_{max}= 1000$)

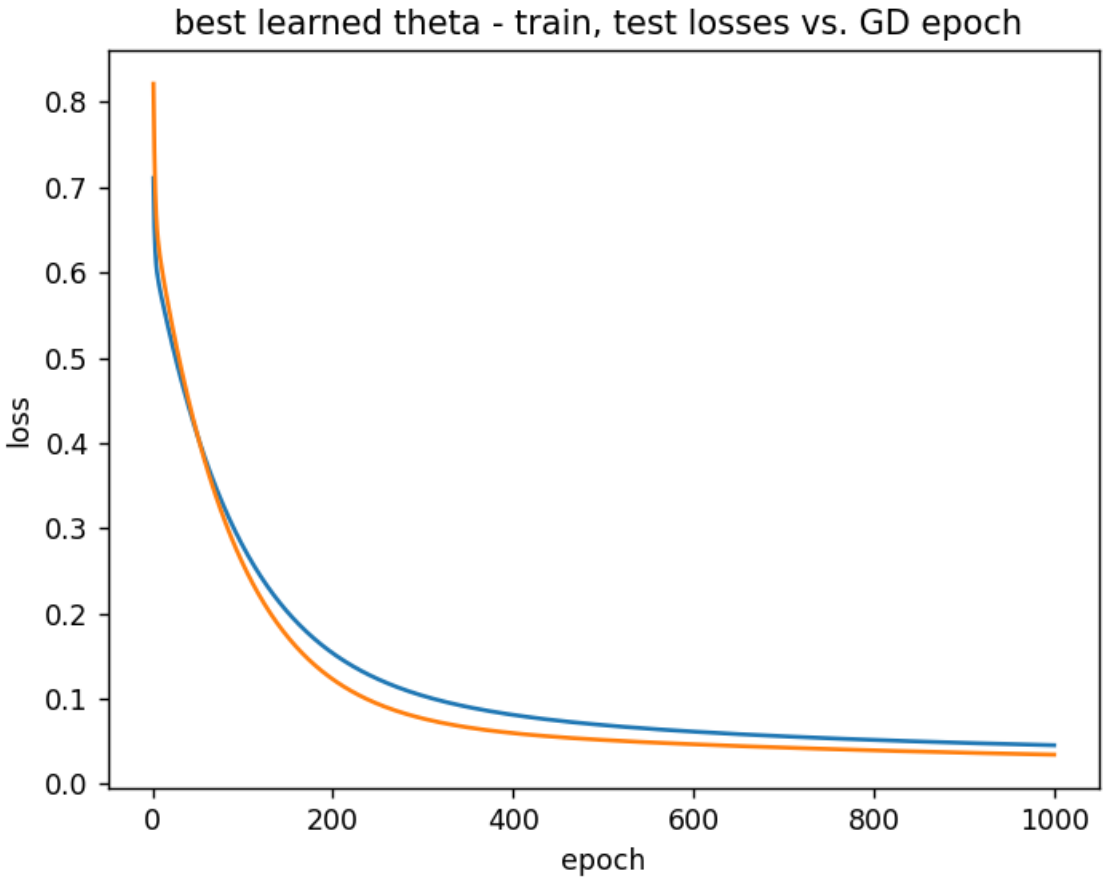

Fig 5: GD epoch vs training loss and test losses. (d=8, t_{max}= 1000)

In Figure 4, there is an additional plot to those generated in Task 3; however, the hyperparameter degree 'd' used in this plot is 6 instead of the best hyperparameter, 8. The difference between Fig 2 and Fig 4 is extremely noticeable with Fig 4 not fitting the dataset as nicely. This is due to the fact that the polynomial degree is lower and thus generalizes the data more than a larger hyperparameter which fits the data better. The worst fitted section in Fig 4 are the beginning points since the curve can't fit that section as well with a lower degree.

In Figure 5, it plots the GD epoch versus the training and testing losses. By using gradient descent optimizations it can be visually seen that the training loss and the gap between the

training loss and testing loss has been reduced. The loss of both the training and testing data is approaching zero as the number of gradient descent epochs increase.

Task 4:

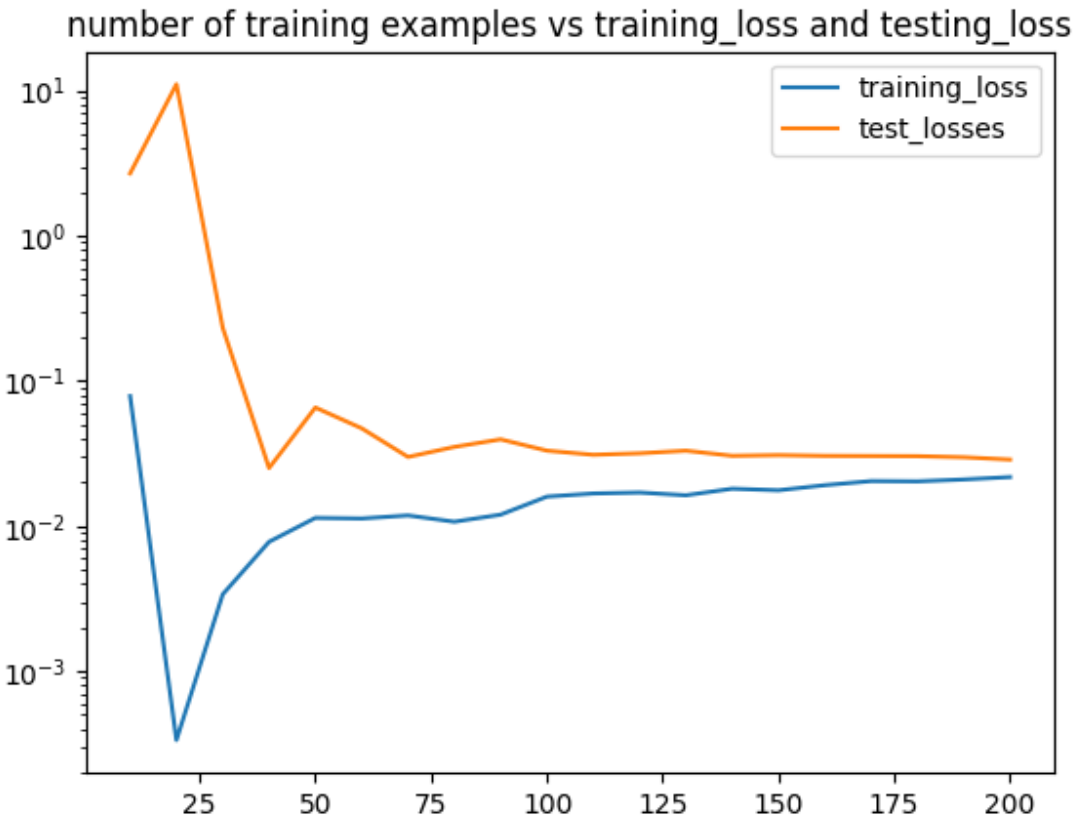

Fig 6: Number of Training Examples vs Training and Testing Loss

In Figure 6, the x-axis represents the value of n, the number of examples used for the model and the y-axis shows the training MSE loss and the test MSE loss. The graph above shows that models that use a small number of examples are more likely to be overfit. An overfit model will have low training error, but have high testing error. This can be seen in the first part of

Figure 6 where $n < 50$. However, as n increases the MSE loss becomes more similar in the training and testing set. The more examples used the better the model will be.

Ridge Regression:

Task 1:

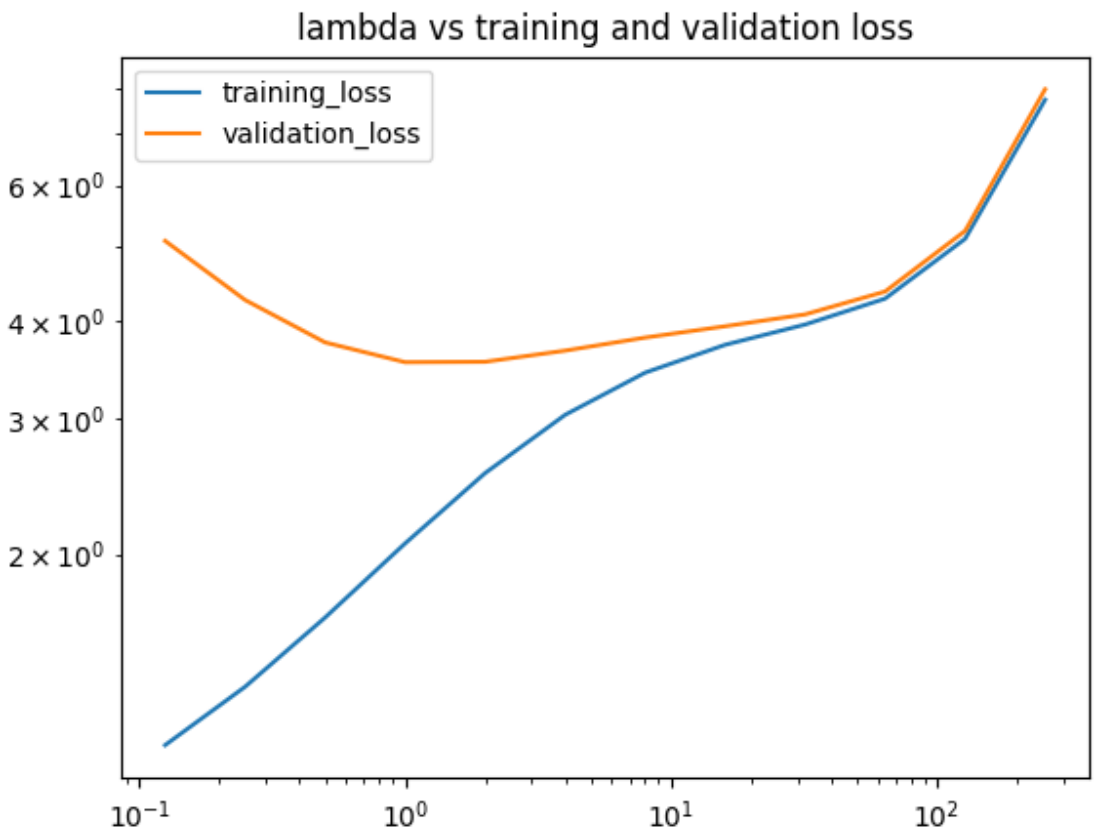

Fig 7: Training Loss and Validation Loss as a Function of Hyperparameter λ

In Figure 7, the y axis is the training loss and validation is the loss, and the x axis is the lambda value. The plot was done using a 4-fold cross validation and thus, the loss is the average loss across the 4 folds during cross validation. In the plot above there is a general trend that as lambda values increase, the training and validation loss also increase. However, the trend of the validation loss is a bit different. With a small lambda value, the validation loss is high and as the lambda value increases the validation loss decreases until it hits the local minimum and then the validation loss begins to grow again when the lambda value is around 10^0. Additionally, the

training loss and validation loss gap continues to decrease as lambda increases. Around lambda = 10^2, validation and training loss are roughly equal. Unlike validation loss, the training loss continues to increase as lambda increases.

Task 2:

The best λ of the previous task was a λ=1.

- The L2 norm of normal beta is 30.269654798800868.
- The L2 norm of best beta is 10.722347108779054
- The L2 norm of large lambda beta is 4.651284674384338.
- The average testing loss for normal beta is 11.031286619642822.
- The average testing loss for best beta is 4.923472034206209.
- The average testing loss for large lambda beta is 12.126420332837405.

A trend can be seen in both the norms and the test loss. In regards to the norms, as the beta values decrease, the L2 norm decreases. The L2 norm calculates the distance of the vector coordinate from the origin of the vector space. This can be seen by the large lambda beta having lower values than both the normal beta and best beta, and the resulting L2 norm being lower than the other L2 norms. As for the average testing loss, the large lambda beta has the largest average and the best beta has the smallest average loss. Both of these trends make sense in regard to Figure 7 shown above. Using cross validation makes the validation loss less noise making us able to pick better hyperparameters.

Task 3:

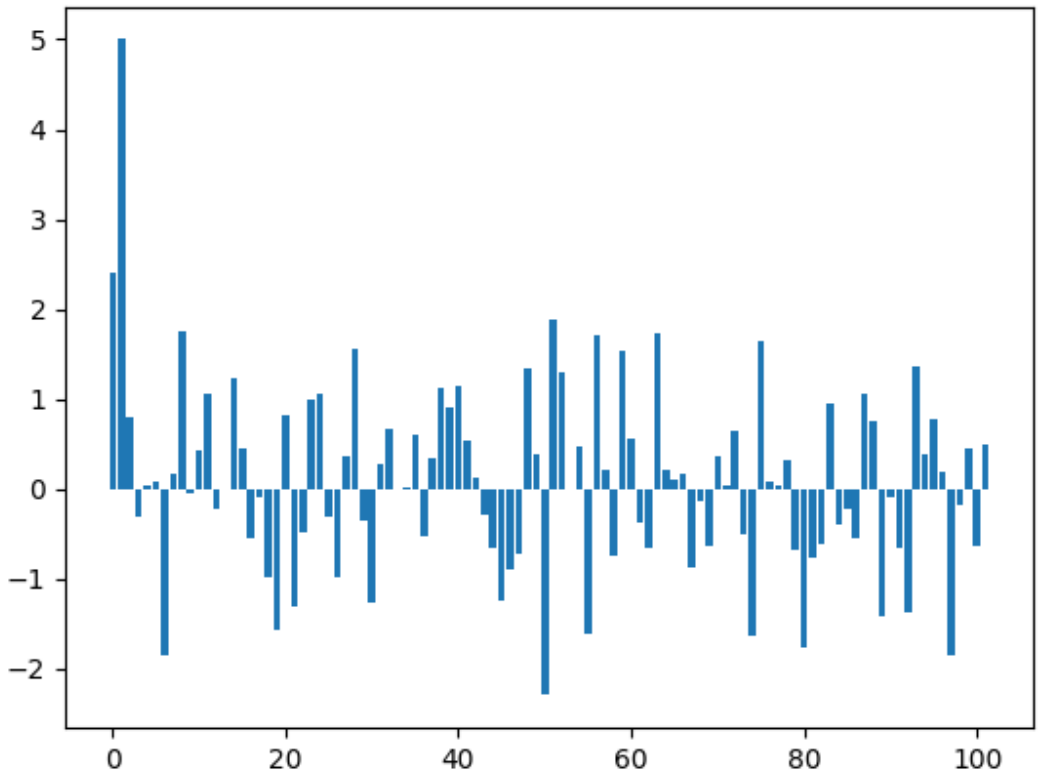

Fig 8: Bar Plot of Learnt Values of Vector From the Best λ

In the Figure above, if β is a p × 1 vector, the x-axis is i where $i \in \{1, \ldots, p\}$ and the y-axis denotes the . The learnt β makes sense in relation to the true underlying distribution since as p increases, β is getting less noise. With larger sets of data, ridge regression works better because of cross validation. Therefore, the bar plot makes sense in relation to the true distribution since as p increases the learnt β becomes less noisy.