

Homework 2

ECGR-4105-001

Joshua Foster

Repo Link: https://github.com/jfoste81/ECGR4105_Homework

Problem 1:

- A) Based on the results in the screenshot below, I would say that the most impactful and important parameter, by far, was the area of the house. The model really clung to this parameter and based most of its guesses off this one feature

```
Final Thetas (Parameters):  
- Intercept: 0.234359  
- area: 837.609534  
- bedrooms: 0.768682  
- bathrooms: 0.369945  
- stories: 0.545732  
- parking: 0.186700
```

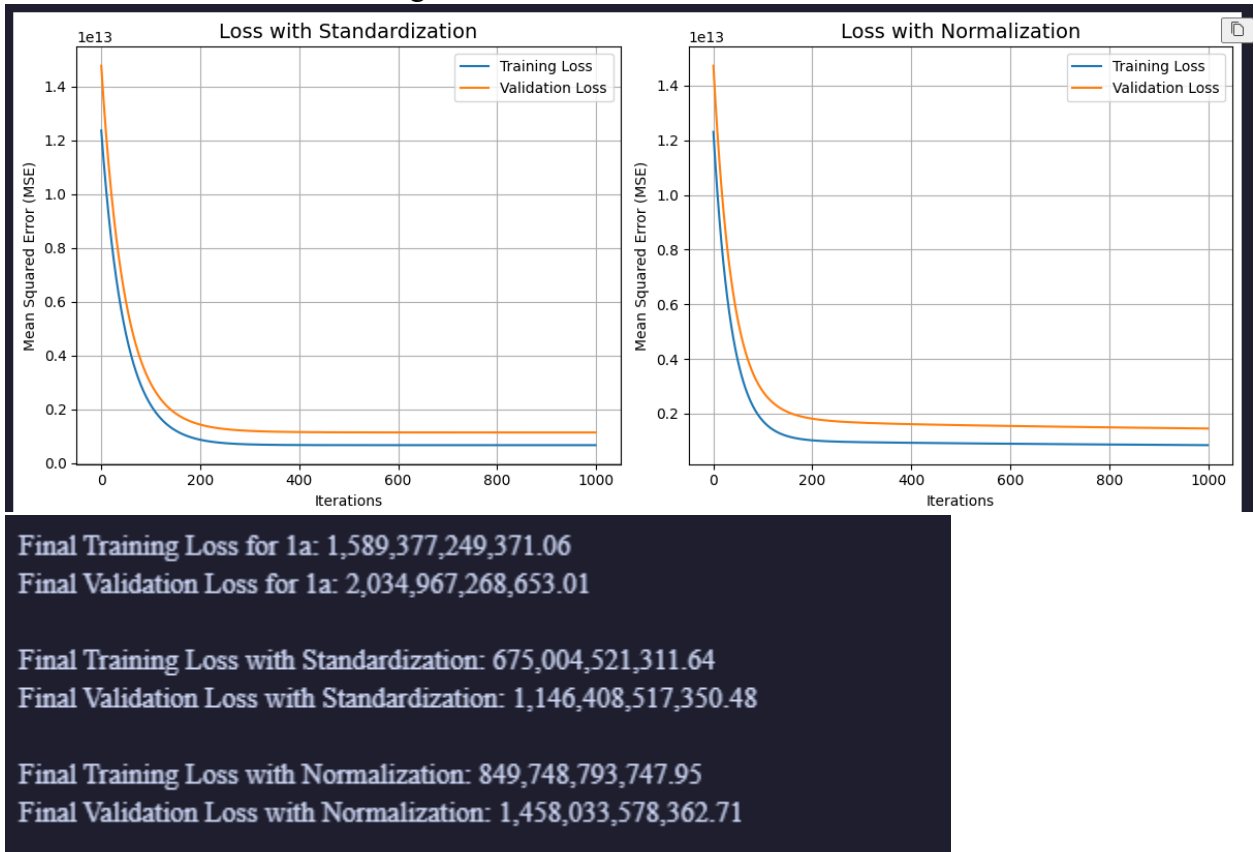
- B) This trend continues in Part B where we see that area still overwhelmingly dominates the other parameters in importance.

```
Final Thetas (Parameters):  
- Intercept: 0.234359  
- area: 837.609487  
- bedrooms: 0.768682  
- bathrooms: 0.369945  
- stories: 0.545732  
- mainroad: 0.206691  
- guestroom: 0.059285  
- basement: 0.119515  
- hotwaterheating: 0.019448  
- airconditioning: 0.127361  
- parking: 0.186700  
- prefarea: 0.075154
```

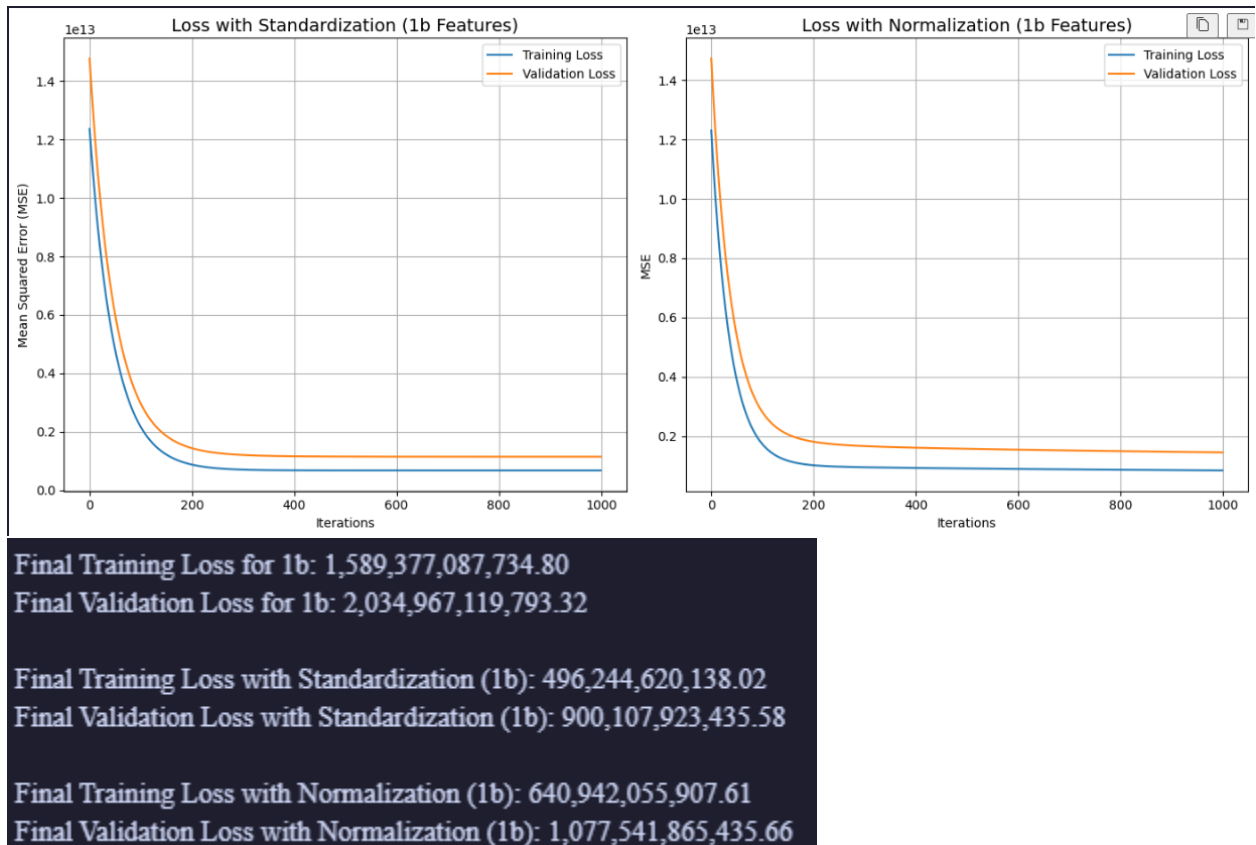
For both of the models in this section, I noted that I had to use a extremely low alpha value ($1e-10$) just to get the model to run. If I tried any alpha higher, it would immediately crash and overflow. All the numbers were too big. Due to this, I had to up the iterations to ~2500 over the usual 1000 iterations. This led to the models completely converging ~2000 iterations into the training.

Problem 2:

- A) With these models, standardization consistently outperformed normalization. As we can see in the screenshots below, both the graph and the numerical prints reflect a small increase in performance in the standardization over the normalization. However, both blow the original loss found in Problem 1 out of the water. This showcases that having one of them is better than having none at all.



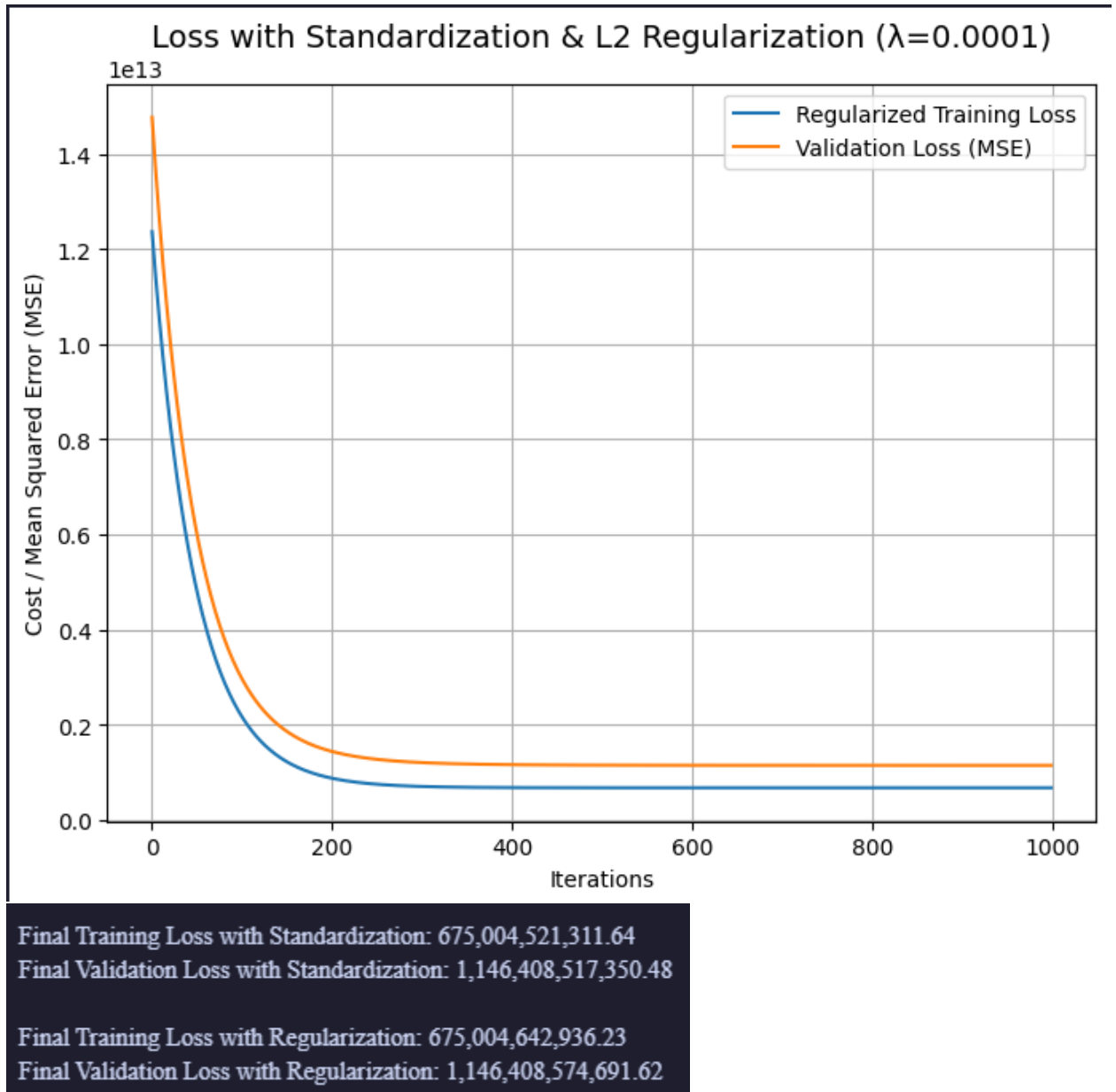
- B) Again, we see that standardization outperforms the normalization in these instances. Both normalization and standardization seem to outperform the losses found in 2A and, consistent with the 2A results, heavily outperform the loss found in 1B.



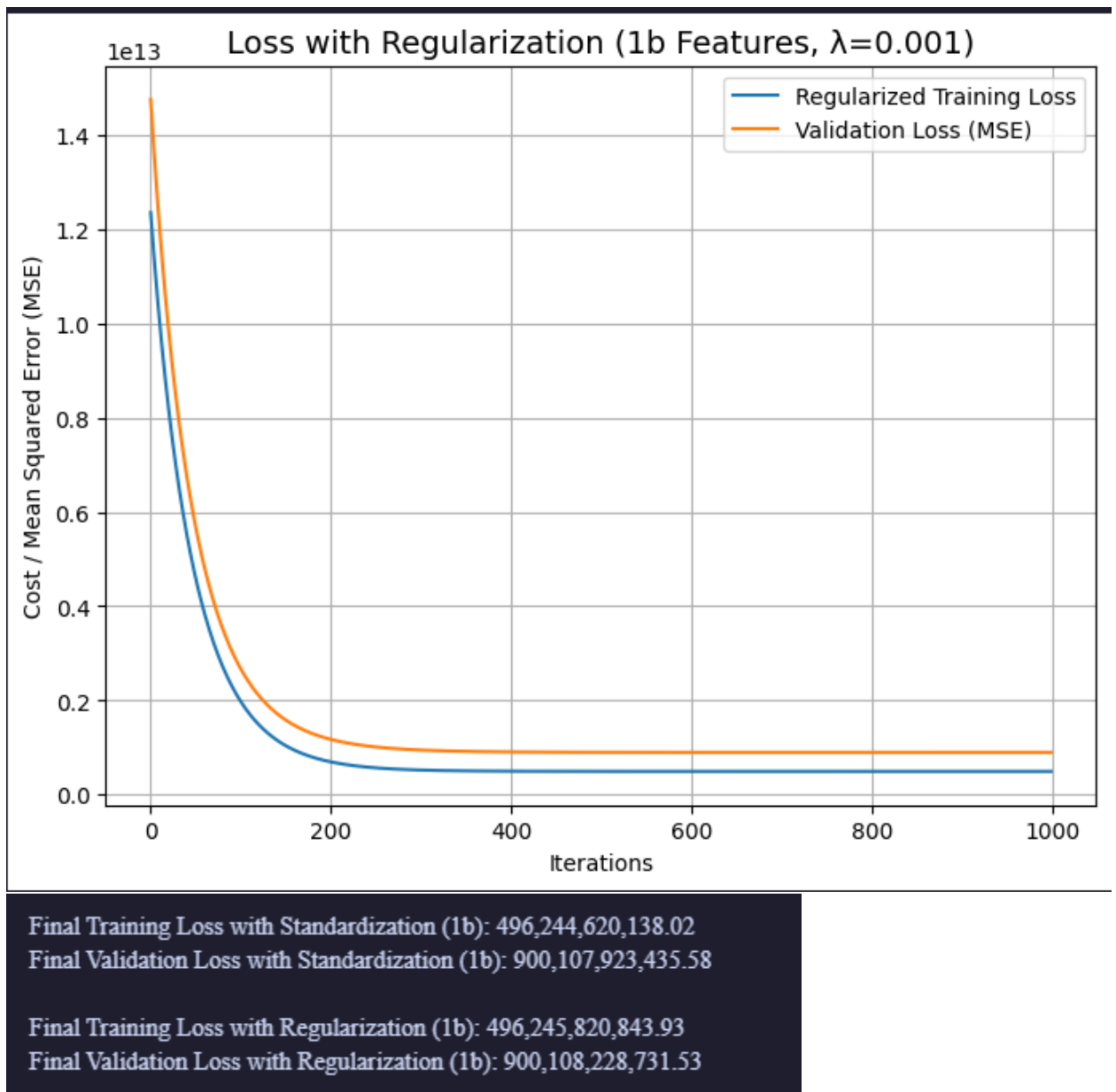
In this problem, I was able to revert to a more normal alpha value of .01, which also enabled me to lower the number of iterations the model trains on (back down to 1000). This model converges around 350 iterations in for standardization, while normalization takes a few hundred more iterations to fully converge.

Problem 3:

- A) My results in this section were not very good, as the regularization did not seem to have any large impact on my results. With a high lambda, the loss was significantly worse than the loss in 2A, but with a very small lambda value the loss was usually slightly lower but sometimes slightly higher than the loss found in 2A.



- B) The same rules applied in this section where the regularization did not seem to positively impact the loss much, if it all. It stayed consistent with higher lambda values hurting the loss compared to 2B, and miniscule values of lambda made small increases and sometimes decreases in model loss.



Honestly, I am not quite sure why my regularization did not help my model more here. I thought it would have more of a positive impact on this section due to the jump in performance we witnessed when applying normalization and standardization to the model. However, the addition of regularization did not seem to help the model much at all, if at all. I experimented heavily with varying values of lambda, but nothing gave me that jump in performance that I was searching for.