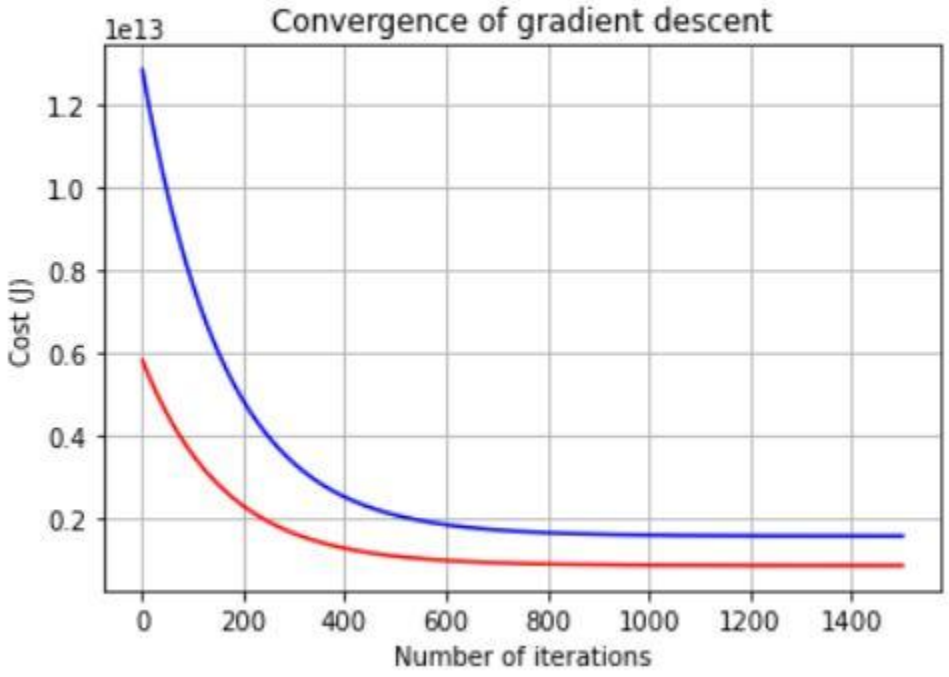
Andrew Hord
801063157
HW1
9/30/21

Github link
https://github.com/AndrewHord555/ECGR-4105-HW0

1.a)Develop a gradient descent training and evaluation code that predicts housing price based on the following input variables:

area, bedrooms, bathrooms, stories, parking,

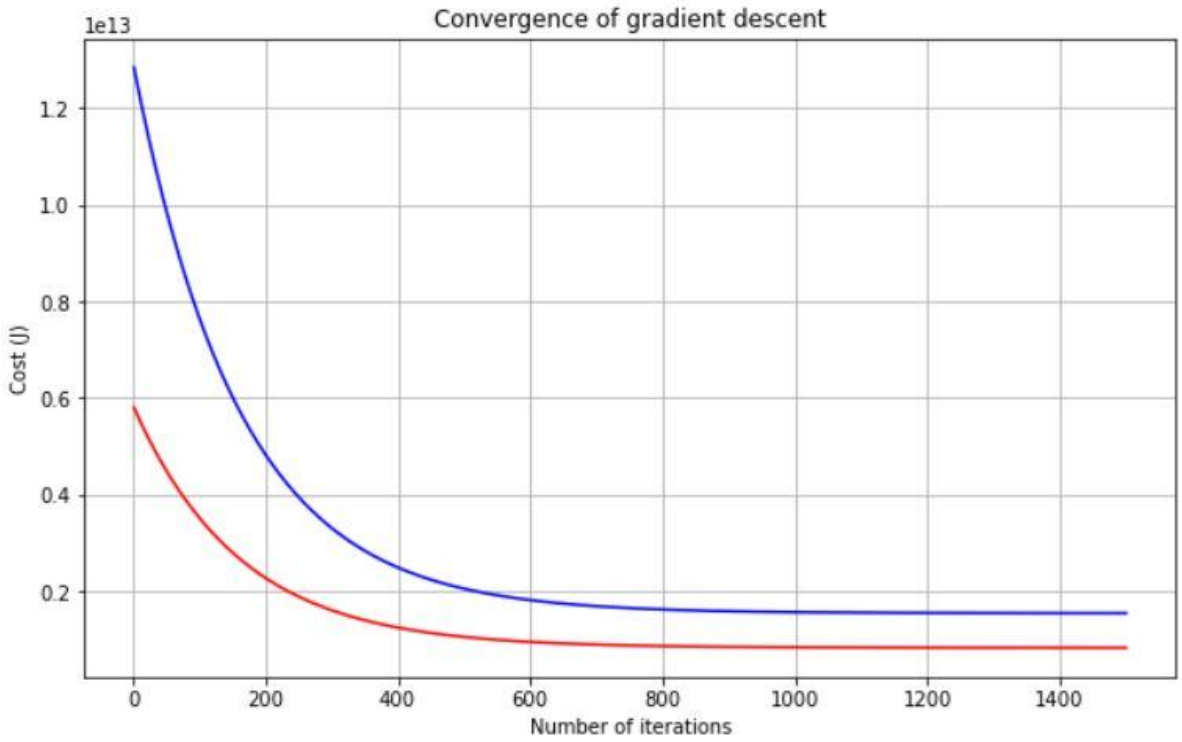
```
Text(0.5, 1.0, 'Convergence of gradient descent')
```




1.b) Develop a gradient descent training and evaluation code that predicts housing price based on the following input variables:

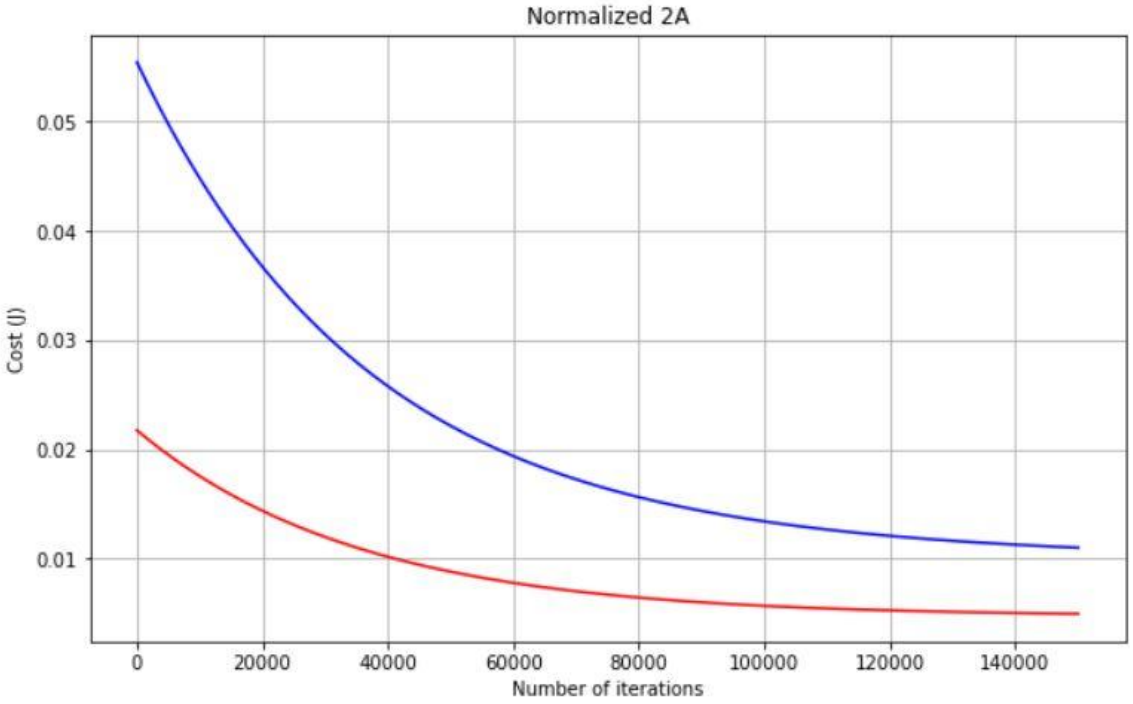
Area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea

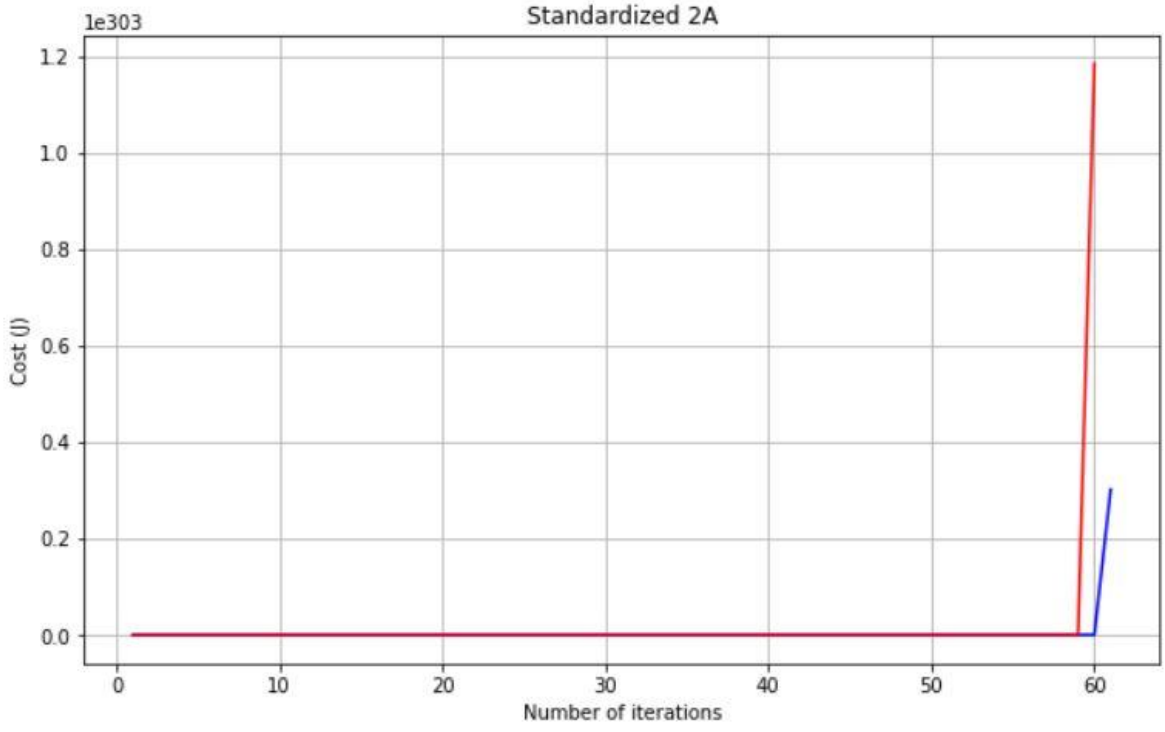
Andrew Hord
801063157
HW1
9/30/21




2.a) Repeat problem 1 a, this time with input normalization and input standardization as part of your pre-processing logic. You need to perform two separate training sessions for standardization and normalization.

Andrew Hord
801063157
HW1
9/30/21

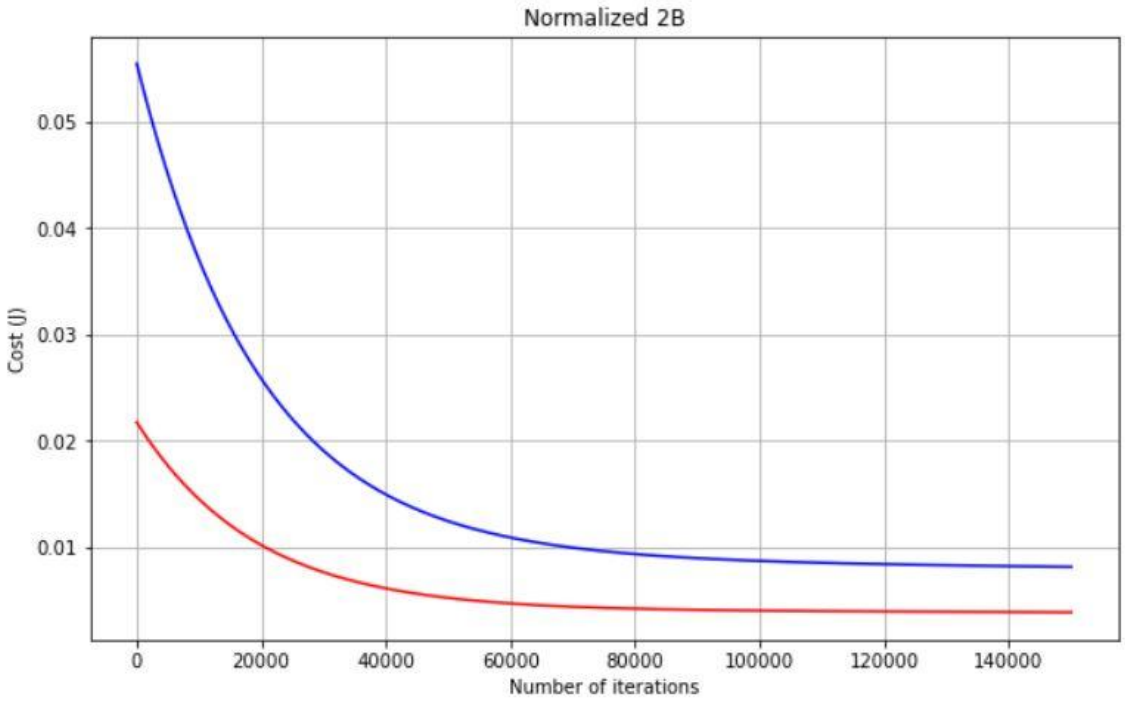


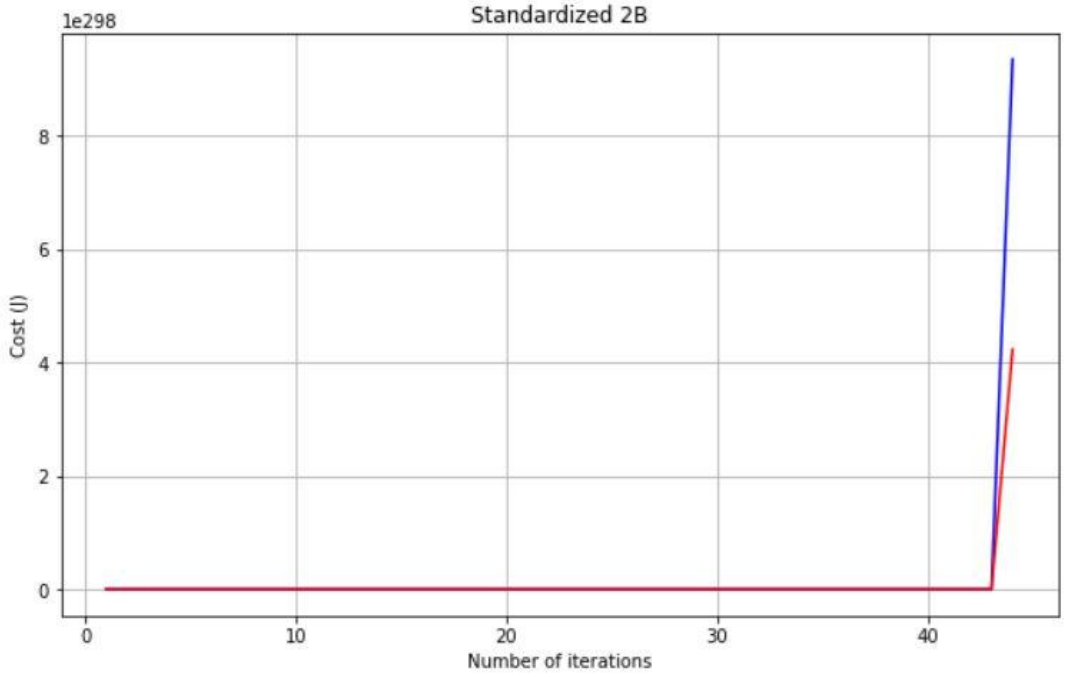
Andrew Hord
801063157
HW1
9/30/21

For problem 2A, the normalization function provides a much better gradient descent when applied to the housing csv file. The normalized data provides a solid gradient descent with a good curve, while the standardized set breaks, by the cost function staying at zero before spiking after a couple dozen iterations has passed. The standardization could have been broken by the standard scaler code being mis applied to the wrong csv file, making the data almost unusable.

2.b) Repeat problem 1 b, this time with input normalization and input standardization as part of your pre-processing logic. You need to perform two separate training sessions for standardization and normalization.



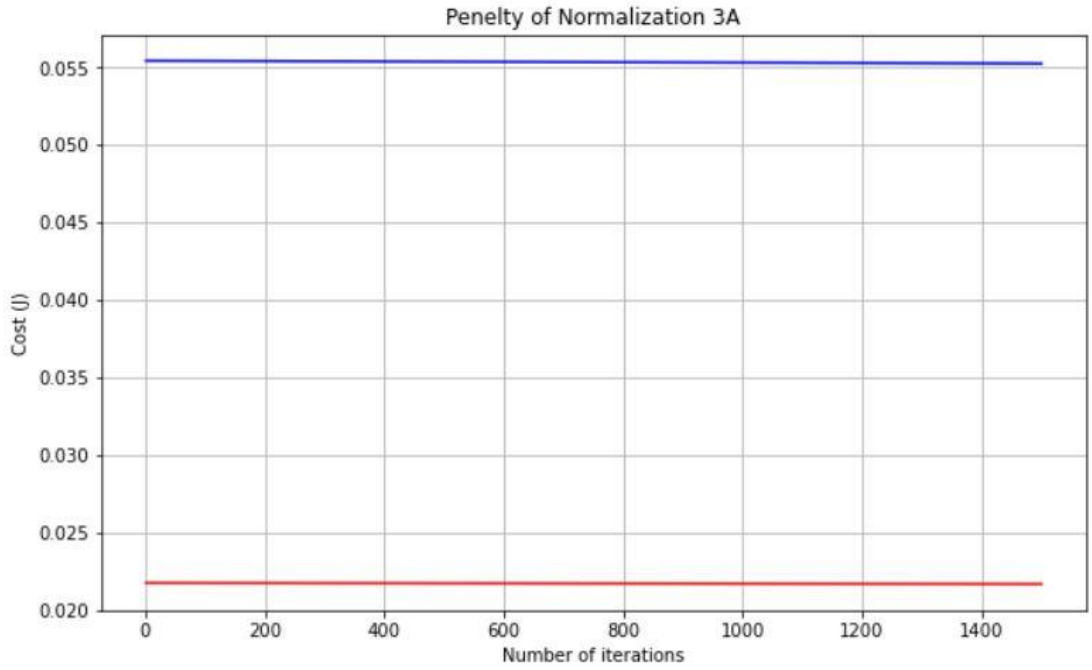
Andrew Hord
801063157
HW1
9/30/21



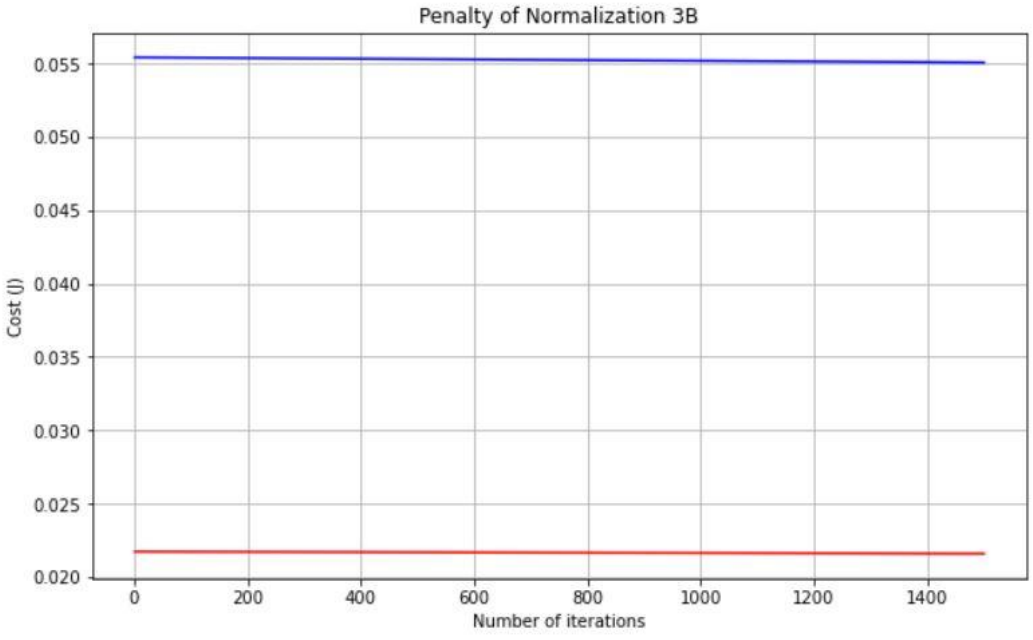

For problem 2B, the normalization function provides a much better gradient descent when applied to the housing csv file. The normalized data provides a solid gradient descent with a good curve, while the standardized set breaks, by the cost function staying at zero before spiking after a couple dozen iterations has passed. The standardization could have been broken by the standard scaler code being mis applied to the wrong csv file, making the data almost unusable.

3.a) Repeat problem 2 a, this time by adding parameters penalty to your loss function. Note that in this case, you need to modify the gradient decent logic for your training set, but you don’t need to change your loss for the evaluation set.

Andrew Hord
801063157
HW1
9/30/21




3.b) Repeat problem 2 b, this time by adding parameters penalty to your loss function. Note that in this case, you need to modify the gradient descent logic for your training set, but you don't need to change your loss for the evaluation set.



Andrew Hord
801063157
HW1
9/30/21
With problem three we had to plot the scaling technique that worked best for us while adding a parameter penalty in our pre-processing logic. When adding the parameter penalties for the normalized data, the gradient descent for both part A and part B, broke where no loss was achieved, no matter what the learning rate ended up being. This likely could have to deal with how the if/else statement for the regularization had been set up causing a variable to be misapplied ruining the regression, or by the normalized data somehow going through the normalized function again, thus causing an issue with the output for part 3.