Name: Long Nguyen
Student ID: 801235507
Homework #: 2

Use the US Housing dataset, develop linear regression with gradient descent algorithm to estimate the price of the house. In this homework, use 80% and 20% split between training and evaluation (test) sets across all problems.

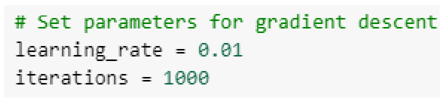
Link to Github repository: https://github.com/lnguye782/ECGR-4105-Intro-to-ML/tree/main/HW2

Problem 1:

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

area, bedrooms, bathrooms, stories, parking

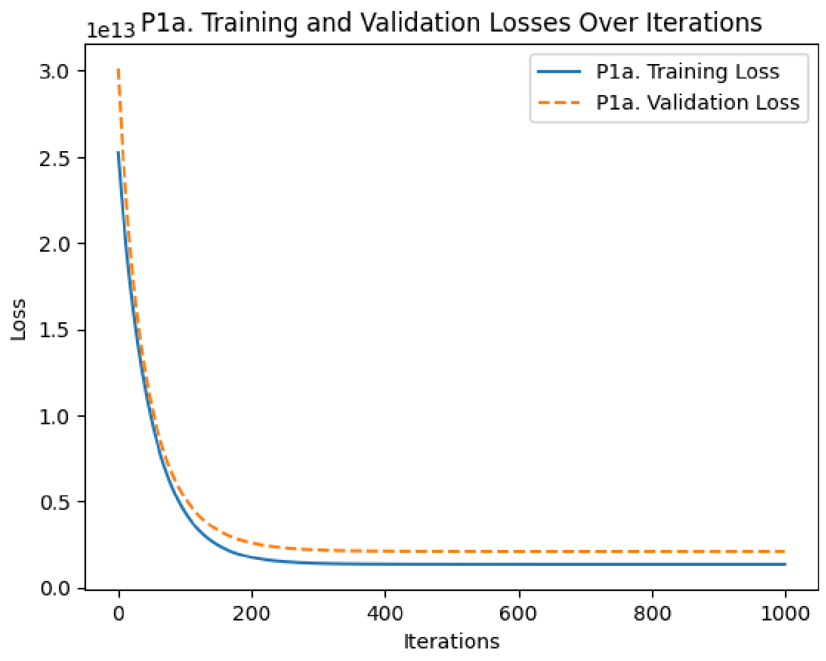
Identify the best parameters for your linear regression model, based on the above input variables.



```
# Set parameters for gradient descent
learning_rate = 0.01
iterations = 1000
```

A learning rate of 0.01 and 1,000 iterations were effective for stability and speed of convergence of loss (wanted the model to reach optimized state)

Plot the training and validation losses (in a single graph, but two different lines). For the learning rate, explore different values between 0.1 and 0.01 (your choice). Initialize your parameters (thetas to zero). For the training iteration, choose what you believe fits the best.



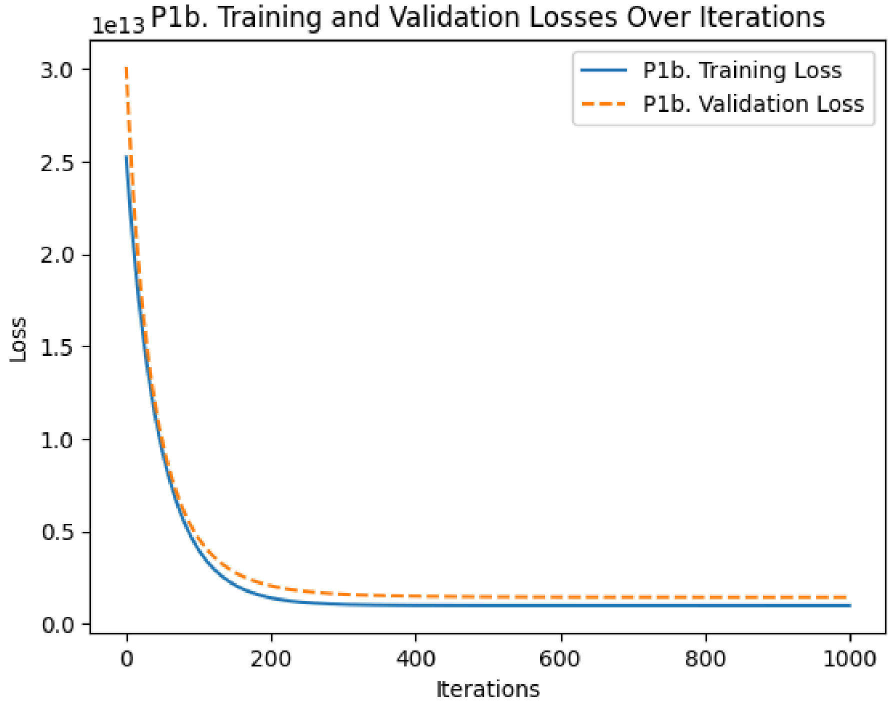
Name: Long Nguyen
Student ID: 801235507
Homework #: 2

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

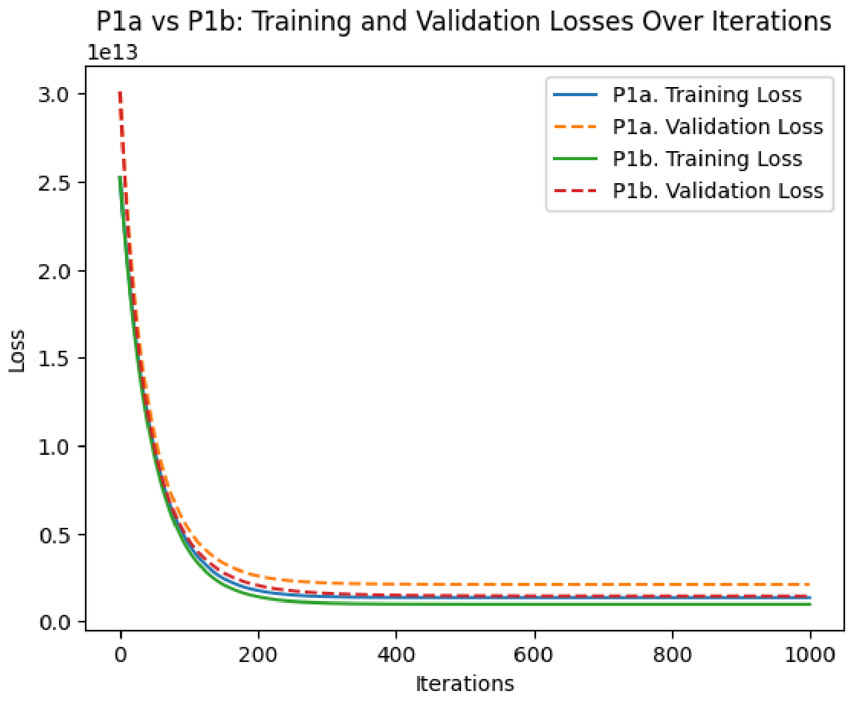
Area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea

Identify the best parameters for your linear regression model, based on the above input variables.

Plot the training and validation losses (in a single graph, but two different lines) over your training iteration. Compare your linear regression model against problem 1 a. For the learning rate, explore different values between 0.1 and 0.01 (your choice). Initialize your parameters (thetas to zero). For the training iteration, choose what you believe fits the best.



Name: Long Nguyen
Student ID: 801235507
Homework #: 2

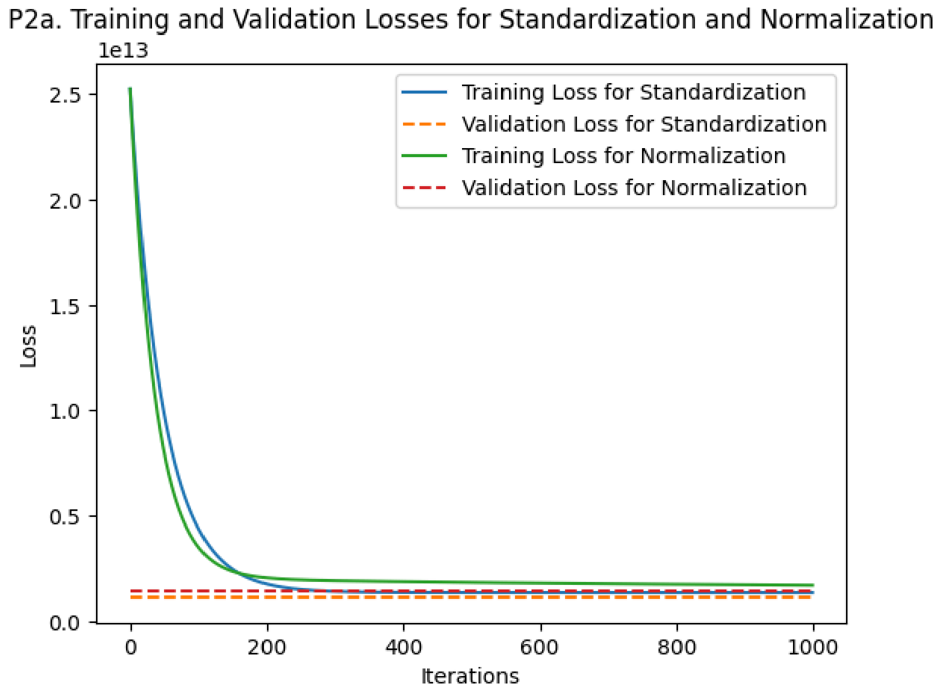


The model in Part 1b shows a lower test mean squared error compared to the model in Part 1a. This tells that by adding multiple input variables, it improves the model's ability to predict housing prices.

Problem 2:

a) Repeat problem 1a, this time with input normalization and input standardization as part of your pre-processing logic. You need to perform two separate trainings for standardization and normalization. In both cases, you do not need to normalize the output!

Plot the training and validation losses for both training and validation set based on input standardization and input normalization. Compare your training accuracy between both scaling approaches as well as the baseline training in problem 1a. Which input scaling achieves the best training? Explain your results.



Both scaling approaches (standardization and normalization) give a similar training loss accuracy and trajectory. The standardization shows to converge slightly faster than normalization.

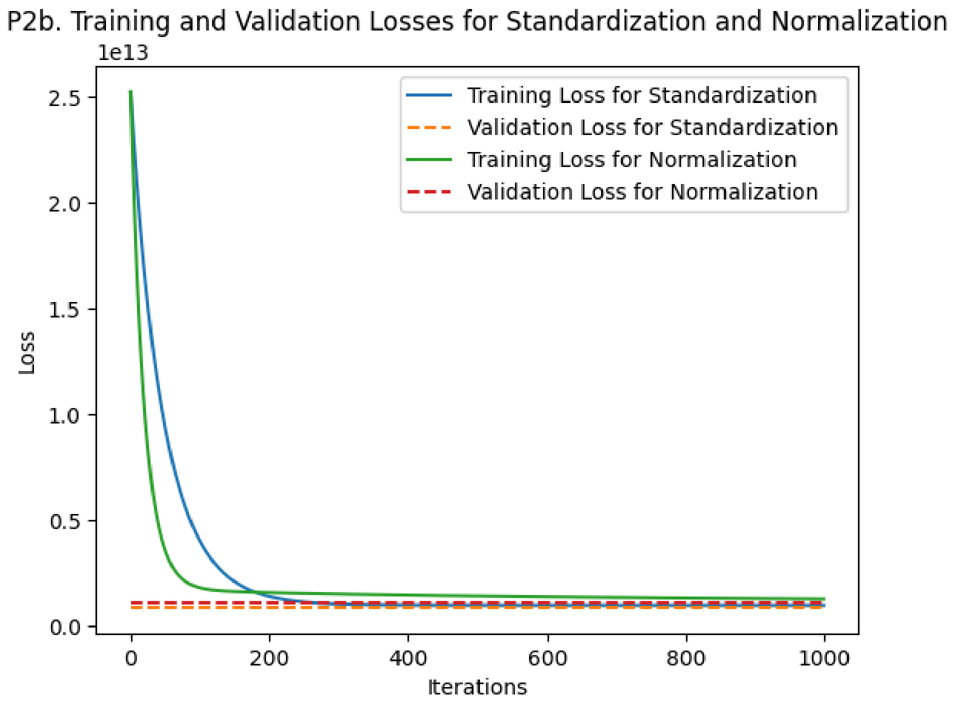
Regarding validation losses, both scaling approaches remain fairly stable (straight lines throughout the iterations). Both validation losses converge to nearly identical values.

The baseline training from problem 1a shows a higher initial loss that converged slower compared to the scaled versions from problem 2a. By applying scaling to input variables, there will be a faster convergence and lower final loss values than the baseline.

Standardization input scaling achieves the best training because of faster convergence (the low-loss plateau reaches earlier than normalization) and lower overall loss.

b) Repeat problem 1b, this time with input normalization and input standardization as part of your pre-processing logic. You need to perform two separate trainings for standardization and normalization. In both cases, you do not need to normalize the output!

Plot the training and validation losses for both training and validation sets based on input standardization and input normalization. Compare your training accuracy between both scaling approaches and the baseline training in problem 1 b. Which input scaling achieves the best training? Explain your results.



The scaling approach of standardization converges faster and reaches the final low loss value more quickly compared to normalization.

Regarding validation losses, both scaling approaches show stable behavior (straight lines throughout the iterations). Both validation losses converge to nearly identical values.

The baseline training from problem 1b shows a slower convergence and higher initial loss compared to the scaled versions from problem 2b. This shows that by applying scaling to input variables, it helps the gradient descent find the optimal solution.

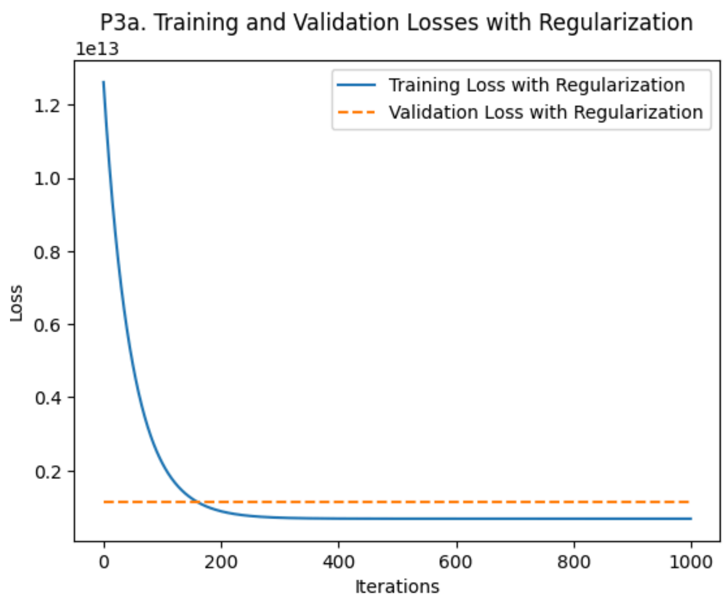
Standardization input scaling still achieves the best training because of faster convergence (the low-loss plateau reaches earlier than normalization) and lower final loss.

Name: Long Nguyen
Student ID: 801235507
Homework #: 2

Problem 3:

a) Repeat problem 2a, this time by adding a 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.

Plot your results (both training and evaluation losses) for the best input scaling approach (standardization or normalization). Explain your results and compare them against problem 2a.



This plot uses the best input scaling approach, which is standardization (explained in problem 2). This result with Regularization shows that there was a smaller gap/more stable between the training and validation losses because Regularization penalizes large values of model parameters (theta) during training (reduces the complexity of the model).

The training loss was higher than in problem 2a because the Regularization term penalizes large parameters and makes the model more conservative.

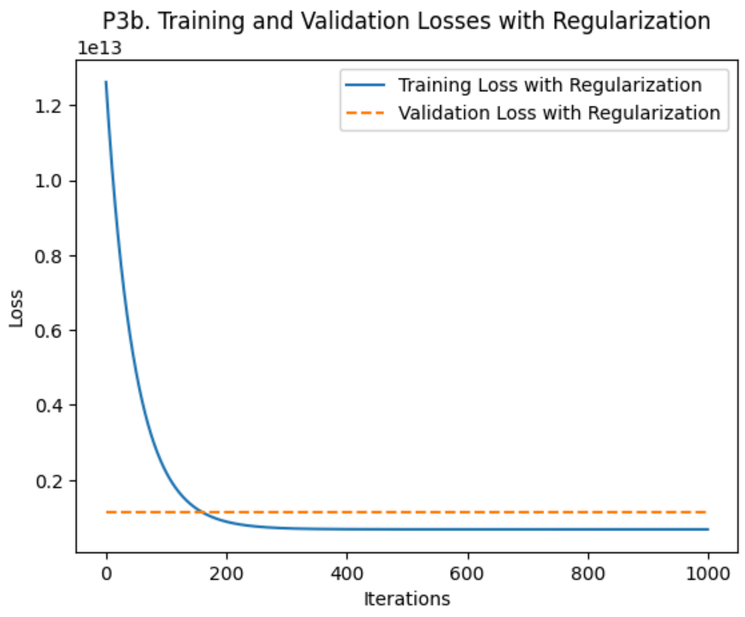
The validation loss shows a slight improvement compared to problem 2a because Regularization helps prevent overfitting. The validation loss shown in problem 3a is lower than problem 3a, which indicates that the Regularization makes the model generalize better.

Overall, by adding a penalty to the parameters via Regularization, a more balanced model was generalized.

Name: Long Nguyen
Student ID: 801235507
Homework #: 2

b) Repeat problem 2b, this time by adding a 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.

Plot your results (both training and evaluation losses) for the best input scaling approach (standardization or normalization). Explain your results and compare them against problem 2b.



This plot uses the best input scaling approach, which is standardization (explained in problem 2). The result without Regularization (in problem 2b) shows that there was a larger gap between the training and validation losses (a sign of overfitting).

The training loss in problem 3b is slightly higher because the model is penalized for large parameter values. This also shows that the Regularization encourages simpler models that do not overfit the training data.

The validation loss in problem 3b is lower and more stable, indicating the overfitting of the model is reduced.

Overall, by adding the Regularization, it helps reduce overfitting. While the training loss increases slightly due to the penalty, the validation loss decreases and remains more stable, indicating better generalization. In contrast, problem 2b (without regularization) likely showed a lower training loss but suffered from higher validation loss, indicating that the model was overfitting.