Brandon Nguyen
COMPE510 - Fall 2025
827813045

Programming Assignment 2 - Linear Regression

A) Results

- **Normal Equations**

```
===== Part 1: Normal Equations =====
First 10 examples from the training dataset:
 x = [32 415 4], y = 32
 x = [16 289 5], y = 51
 x = [41 402 4], y = 35
 x = [40 618 2], y = 40
 x = [21 513 4], y = 42
 x = [13 383 7], y = 42
 x = [33 204 8], y = 48
 x = [2 184 6], y = 45
 x = [30 769 7], y = 25
 x = [33 87 10], y = 47

beta computed from the normal equations:
 44.180059
 -0.248674
 -0.005617
 1.227765

Predicted price of the house: 50.70
Program paused. Press enter to continue.
```

- **Evaluate Performance**

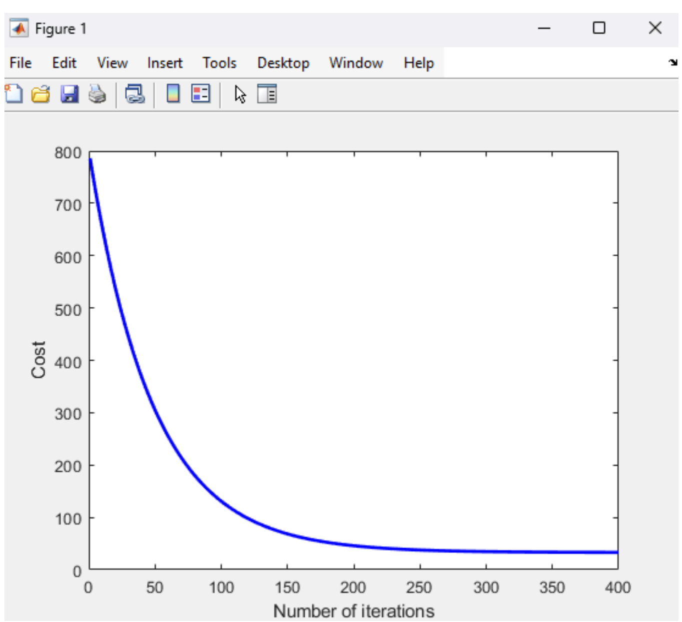
```
===== Part 2: Evaluate Performance =====
Average prediction error (using normal equations):
 76.670451

Predicted price of the house (using normal equations):
 $50.696353

Program paused. Press enter to continue.
```

- **Feature Normalization**

```
===== Part 3: Feature Normalization =====
Normalizing Features ...
beta computed from the normal equations after feature normalization:
 38.250345
 -2.827541
 -7.136066
 3.566468

Average prediction error after normalization (using normal equations):
 76.670451

Program paused. Press enter to continue.
```

- **Gradient Descent**

```
===== Part 4: Gradient Descent =====
beta computed from gradient descent:
 37.563730
 -2.803395
 -6.774042
 3.913907

Average prediction error (using gradient descent):
 73.421086

Predicted price of the house (using gradient descent):
 $50.285527
>>
```

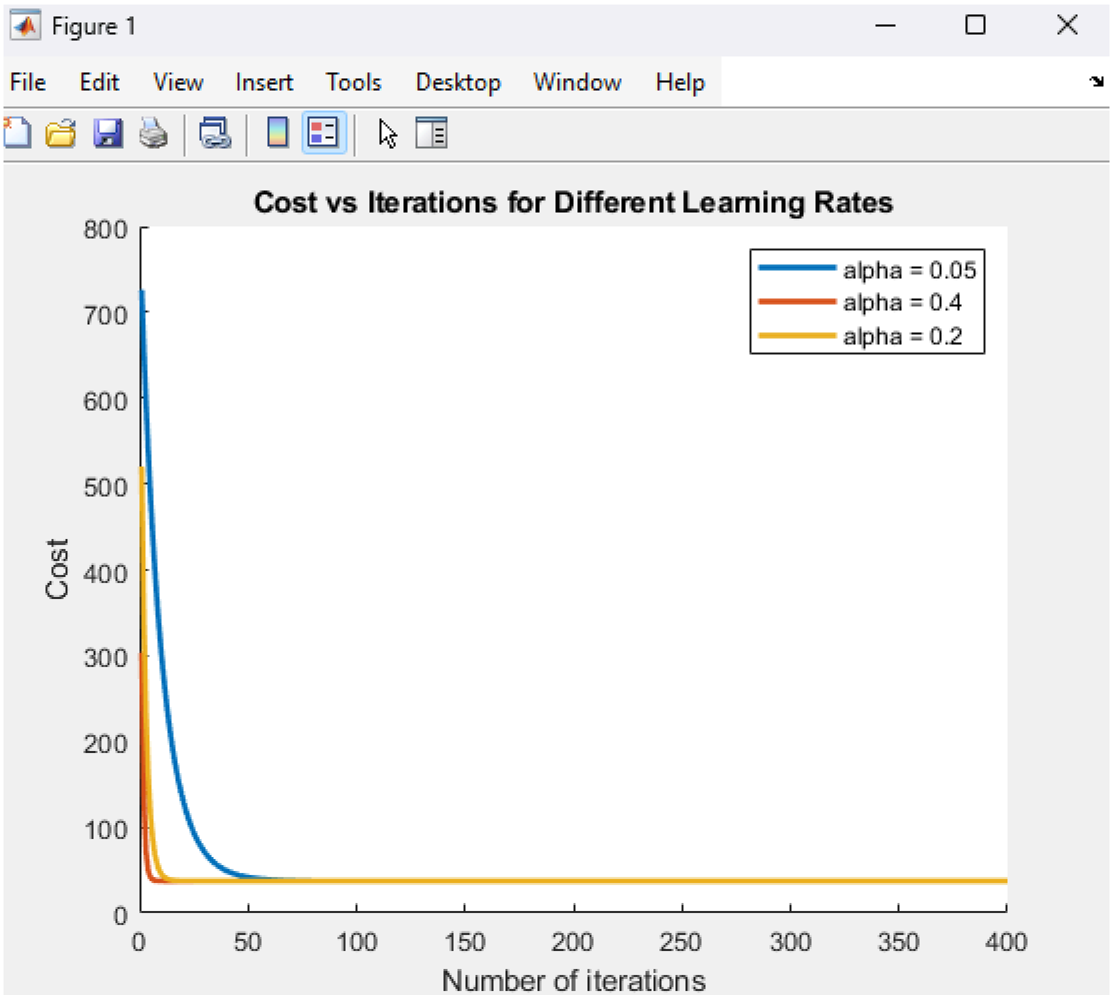


B) Questions

- **Run gradient descent with at least three different learning rates. Show how the cost function behaves over iterations. Based on the results, explain how the learning rate affects convergence and model performance. What learning rate worked best for your data, and why?**

After running gradient descent with at least three different learning rates, the predicted price and prediction error seem to be the same across the board. However, looking at the graph, some conclusions we can draw is that smaller values converge slower and bigger values converge faster. Provided below will be the output testing gradient descent with at least three different learning rates of 0.05, 0.4, and 0.2.

```
===== Part 4: Gradient Descent =====
For alpha = 0.050000
beta computed from gradient descent:
 38.311034
 -2.552807
 -6.222839
 4.077584
Average prediction error: 104.742275
Predicted price of the house: $50.138935

For alpha = 0.400000
beta computed from gradient descent:
 38.311034
 -2.552765
 -6.223262
 4.077160
Average prediction error: 104.741975
Predicted price of the house: $50.138505

For alpha = 0.200000
beta computed from gradient descent:
 38.311034
 -2.552765
 -6.223262
 4.077160
Average prediction error: 104.741975
Predicted price of the house: $50.138505

Predicted price of the house (using gradient descent):
 $50.138505
>>
```



- **What is the role of feature normalization in gradient descent? What would happen if you applied normalization incorrectly - for example, normalizing the training set but not the test set?**

The feature normalization file called in the gradient descent file plays a crucial role because normalization helps the gradient descent run smoothly and efficiently because normalization helps with timing speed and balances values being tested.

- **Which method (normal equation or gradient descent) gave better test performance? Why do you think that happened?**

From looking at my output results above between the normal equation and gradient descent files, it seems like the gradient descent program gave a better test performance for that run because it had a lower error rate. This is most likely because gradient descent is more accurate, and was able to run more smoothly because of factors like timing and better testing than the normal equation.

- **Suppose you had access to more input features (e.g., floor level, building material, or school district). How would adding those affect your current implementation and predictions?**

Adding more input features like floor level, building material, or school district would definitely change up the program implementation and predictions because there would be more variables to take into consideration. For example, a floor level category may work because it has some numbers, but building material would not because it is words and the school district category would also be words or long numbers, messing up the calculations or programs. In general, is it clear that adding multiple new factors will clearly mess with the predictions, calculations, and implementation of the program.

C) Summary

In this second programming assignment, I believe that the implementation of each of the program files went well in connection to the main program file. The things that went well regarding this second assignment include producing outputs for the instructions/requirements, and learning more about machine learning concepts like gradient descent and how it can be applied in Matlab programming. On the other hand, some challenges along the way included debugging my code when it did not produce an output at all, and fixing my code at first when it was producing the wrong output from what was expected. Overall, this second assignment was efficient in teaching how to use more machine learning topics like gradient descent and how to import a datasheet into Matlab to use data in a program and calculations.