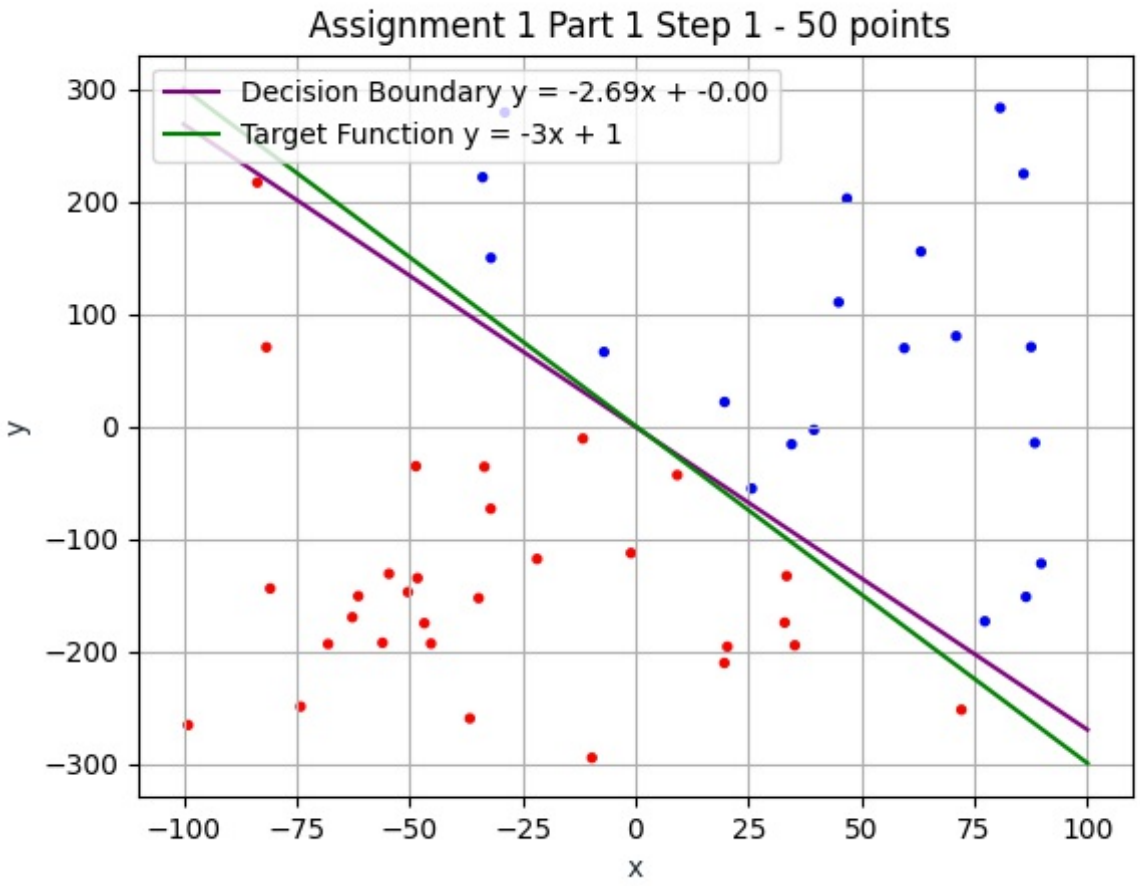
# CMPE 462 Assignment 1 Report

## Part 1

- Samples are randomly chosen from [-100,100) range for x and [-300,300) range for y. Range of y is 3 times that of x, in order to make sure the final plot looks full while the target function plot line is fully visible for that range.
- Random seed can be set by changing the global variable RANDOM_SEED By default it is kept as None (numpy requests a random seed from os)
- Below are the results and plots for each step with RANDOM_SEED=12345

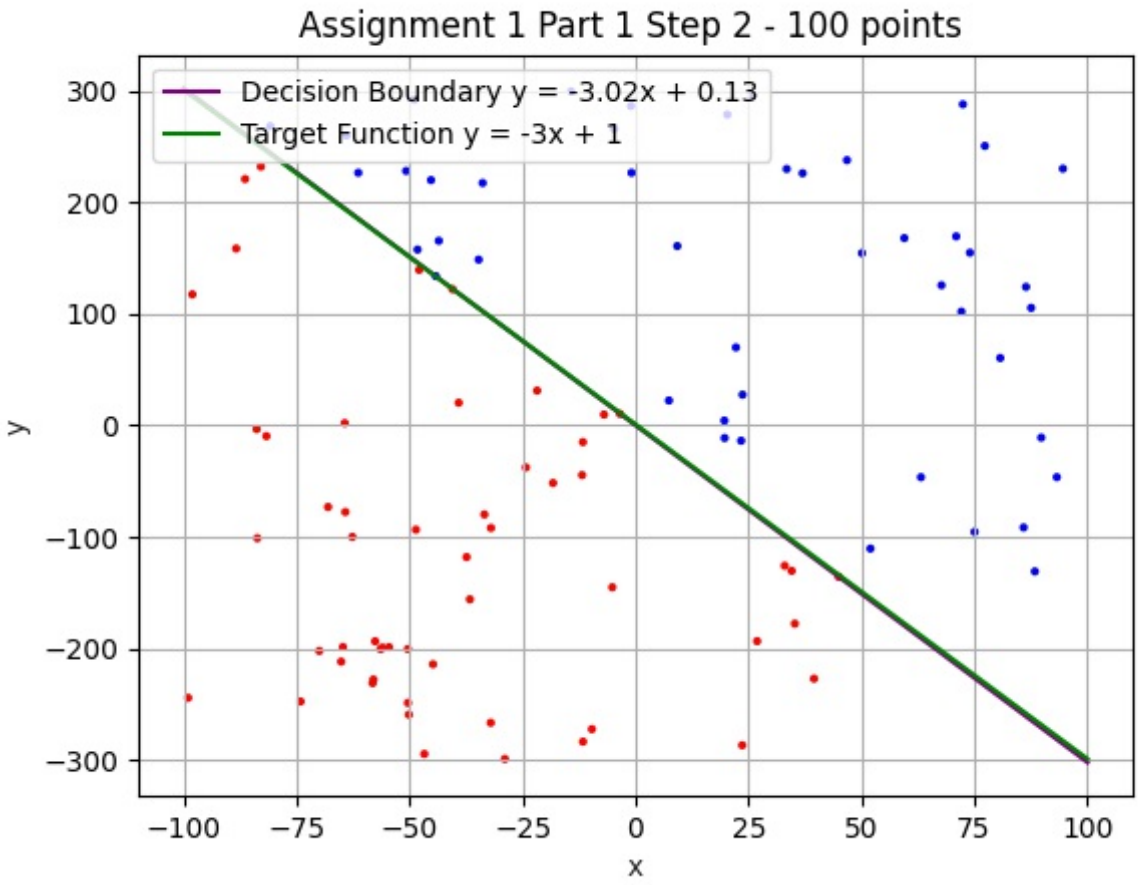
Step 1

```
Finished calculation in 25 iterations
Calculated weights are: [  0.         531.02365676 197.25030511]
Decision Boundary is y = -2.69x + -0.00
```



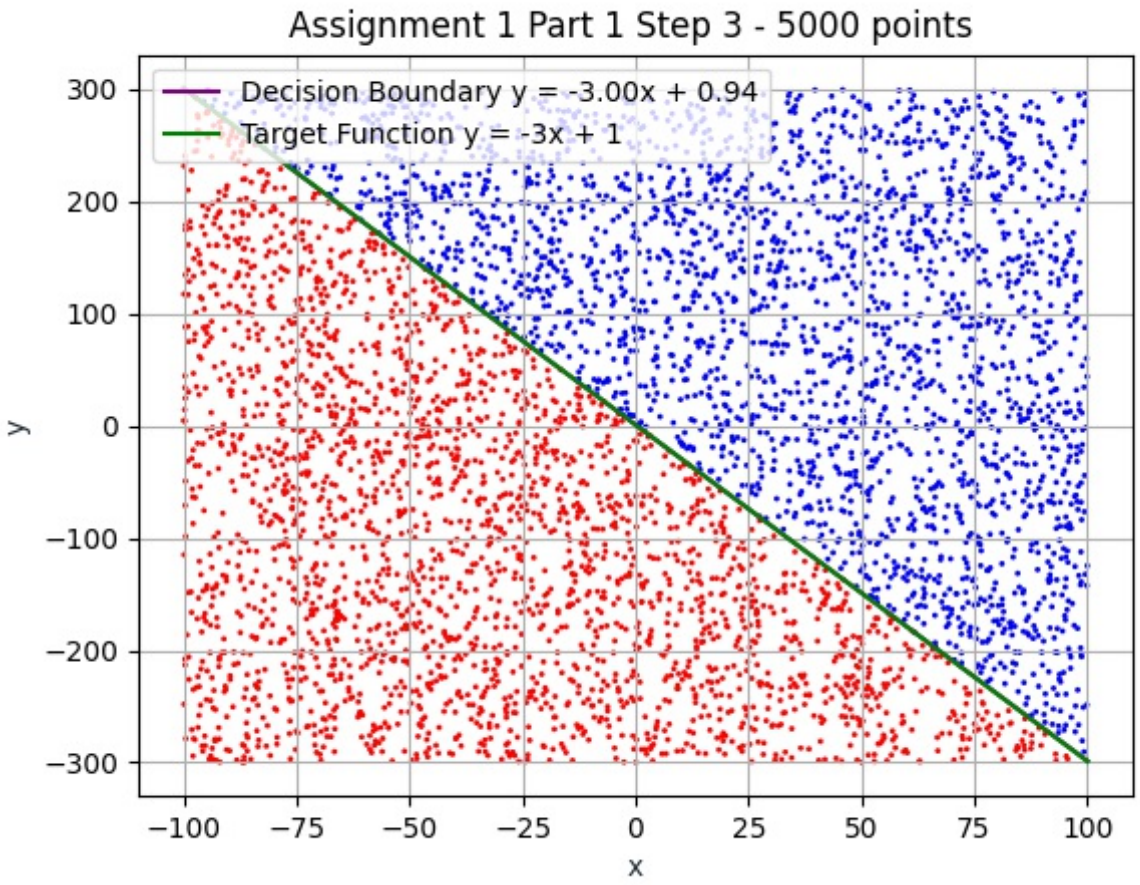

Step 2

```
Finished calculation in 119 iterations
Calculated weights are: [ -50.         1122.35262344  371.32963463]
Decision Boundary is y = -3.02x + 0.13
```




Step 3

```
Finished calculation in 23867 iterations
Calculated weights are: [-6418.         20547.05277251  6856.03386458]
Decision Boundary is y = -3.00x + 0.94
```




Part 1 Overview

When we check out above results, we can see
that the decision boundary gets closer to the original target function
as the number of sampled points are increased.

Also, as the number of sampled points are increased,
PLA needs to make much more iterations to fully converge.

See below table for comparing above results.

| Step | Samples | Iterations | y = -3x + 1 |
|---|---|---|---|
| 1 | 50 | 25 | y = -2.69x + -0.00 |
| 2 | 100 | 119 | y = -3.02x + 0.13 |
| 3 | 5000 | 23867 | y = -3.00x + 0.94 |

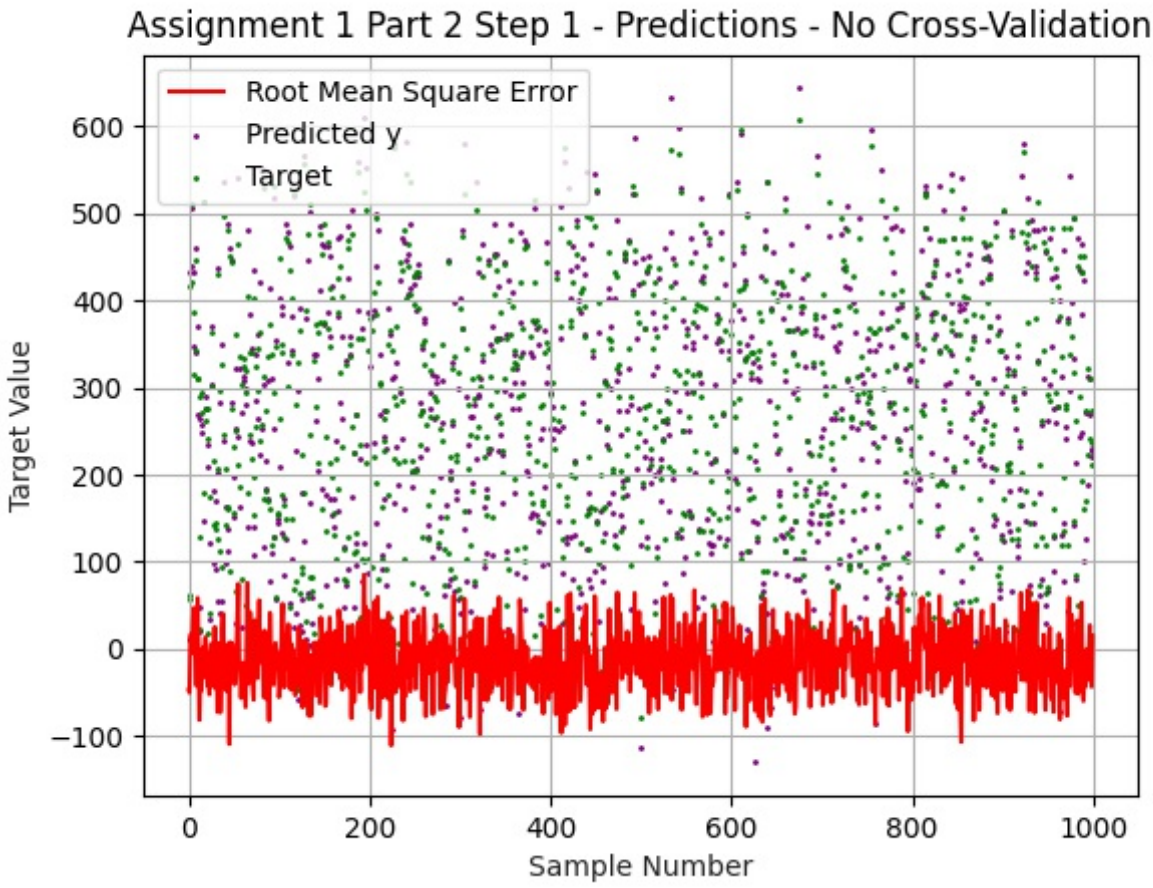
## Part 2

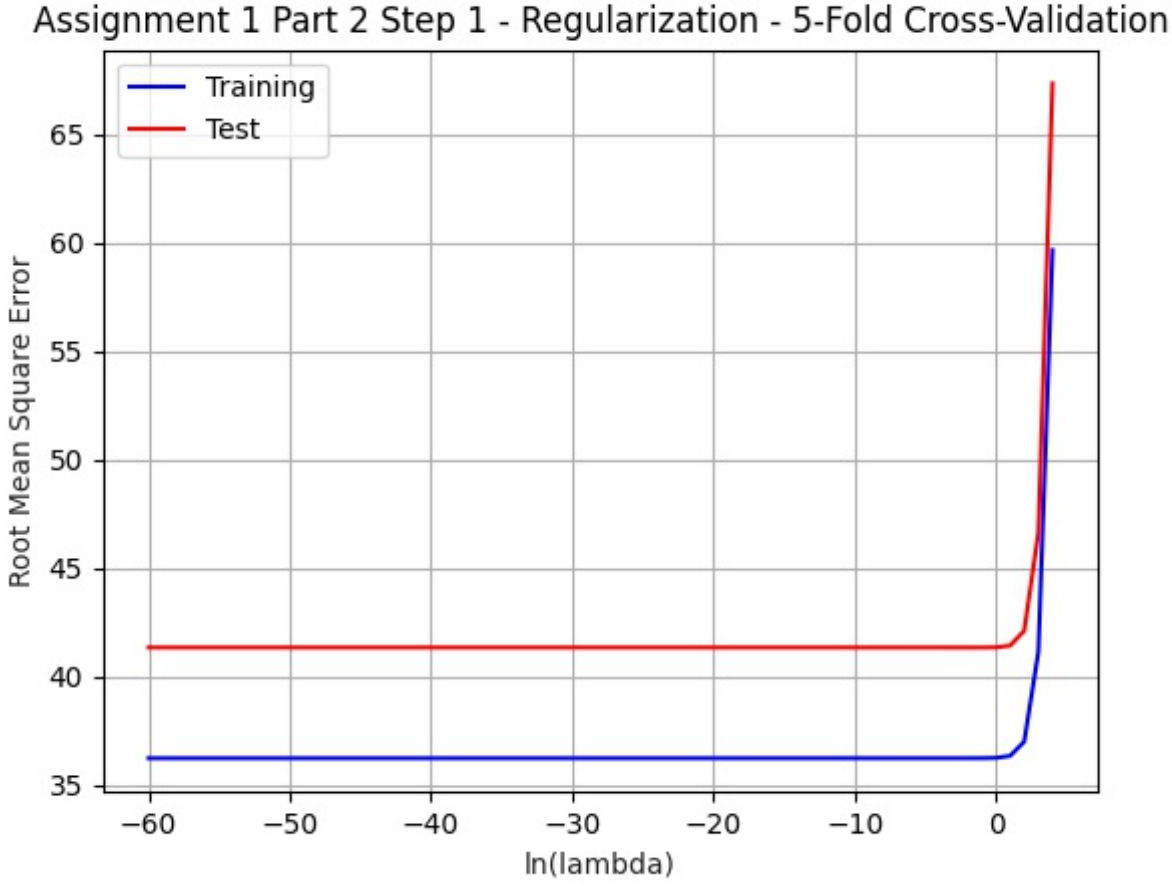
- Implemented closed form solution, so there will not be any loss over iterations graphs.
- Chose to apply 5-fold cross validation,
  this can be changed via the global variable, PART2_S_FOLD_S_VALUE
- Created a few supplementary graphs in order to get a better grasp of how well it predicts, and which lambda value to choose.

- Timing includes cross validation, but none of the extra-supplementary graph generations.
- Set PART2_PLOT_PREDICTIONS=True to create part2_stepN_predictions.png which shows predictions vs actual target values for each sample.
- Set PART2_PLOT_LAMBDA_VALUES=True to create part2_stepN_regularization.png a graph of training & testing results (root mean square errors) over varying ln-lambda value. This will be same for step2 and step3 as expected, because they are only varied by regularization.
- Chosen the lambda value for regularization as e^-10 yet it is neither beneficial nor harmful, as it seems.
- When we choose to do 2-fold cross validation instead, regularization becomes useful because of tolerating the lack of many training values causing over-fitting. But such analysis is out of this report's scope. If interested, one can set PART2_S_FOLD_S_VALUE=2 and PART2_PLOT_LAMBDA_VALUES=True and run the program on part2, and check out the generated graph, part2_stepN_regularization.png.

Step 1

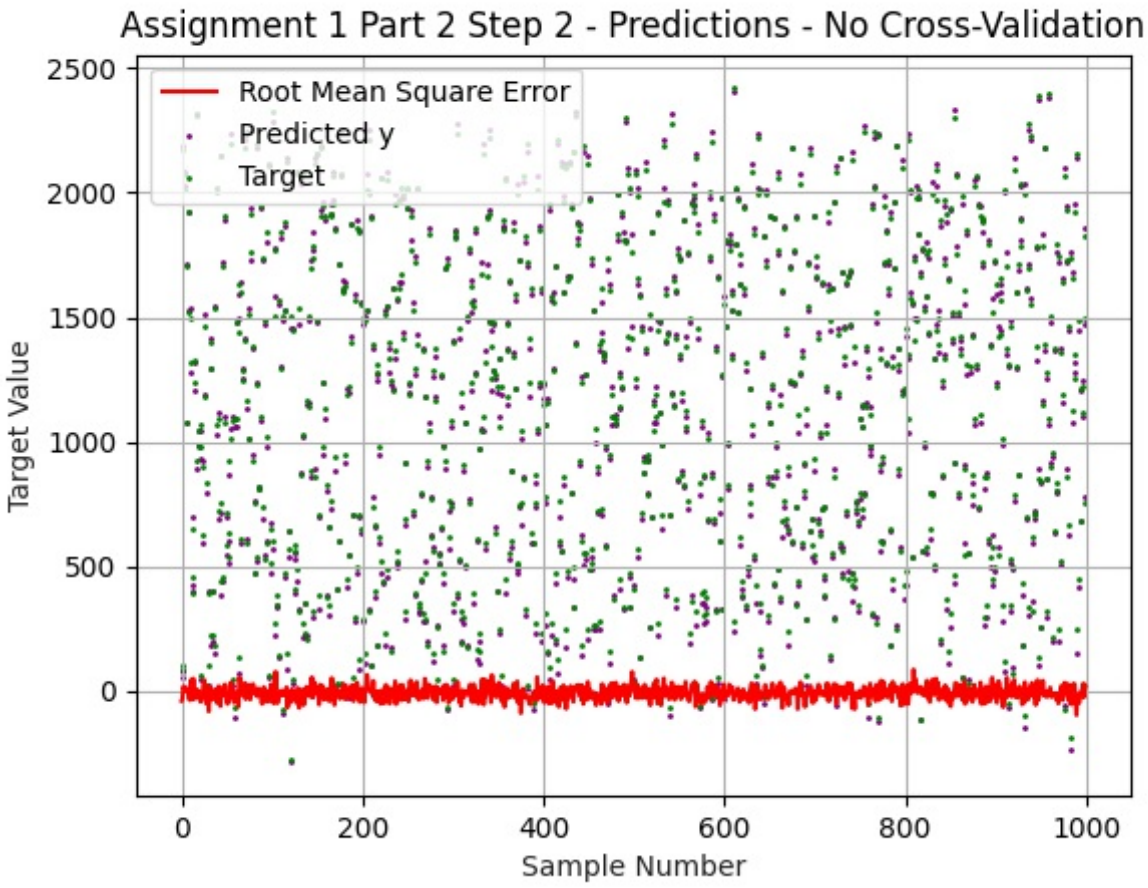
```
Applying 5-fold cross validation
There were 100 independent variables and 1 dependent variables
There were 1000 samples in total
Completed in 3.160 milliseconds
Average Erms_train = 36.257771814312136
Average Erms_test = 41.3584787667631
```

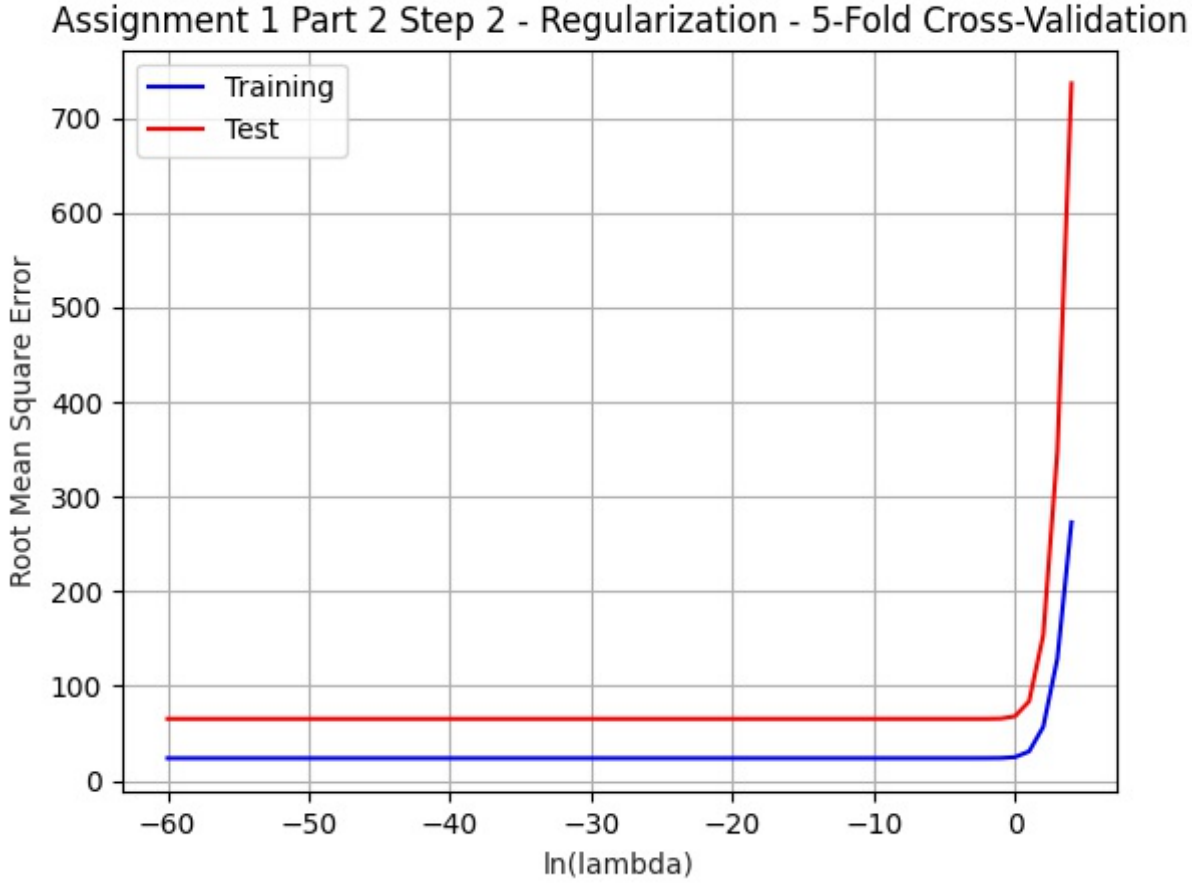

Assignment 1 Part 2 Step 1 - Predictions - No Cross-Validation
Root Mean Square Error
Predicted y
Target
Target Value
Sample Number
600
500
400
300
200
100
0
−100
0
200
400
600
800
1000


Assignment 1 Part 2 Step 1 - Regularization - 5-Fold Cross-Validation
Training
Test
Root Mean Square Error
ln(lambda)
65
60
55
50
45
40
35
−60
−50
−40
−30
−20
−10
0

Step 2

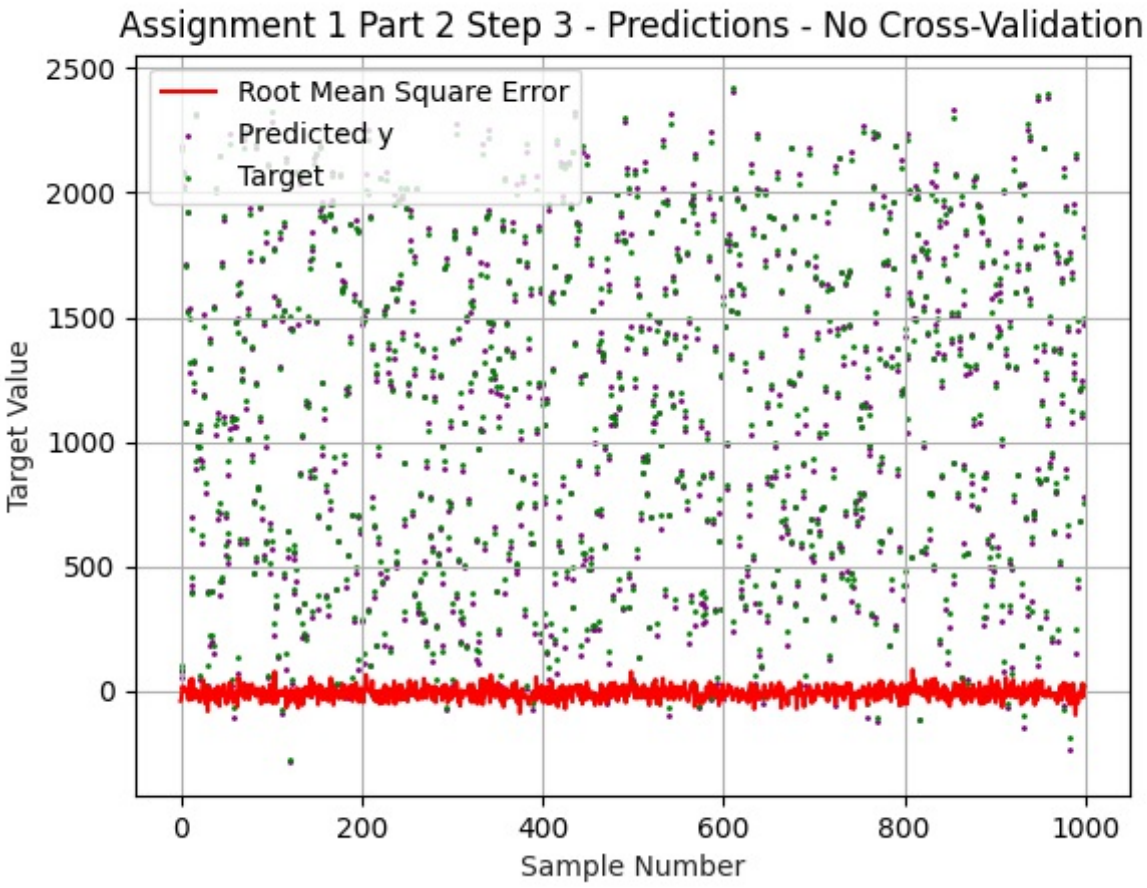
```
Applying 5-fold cross validation
There were 500 independent variables and 1 dependent variables
There were 1000 samples in total
Completed in 48.580 milliseconds
Average Erms_train = 23.6657974544233
Average Erms_test = 65.28337321386573
```

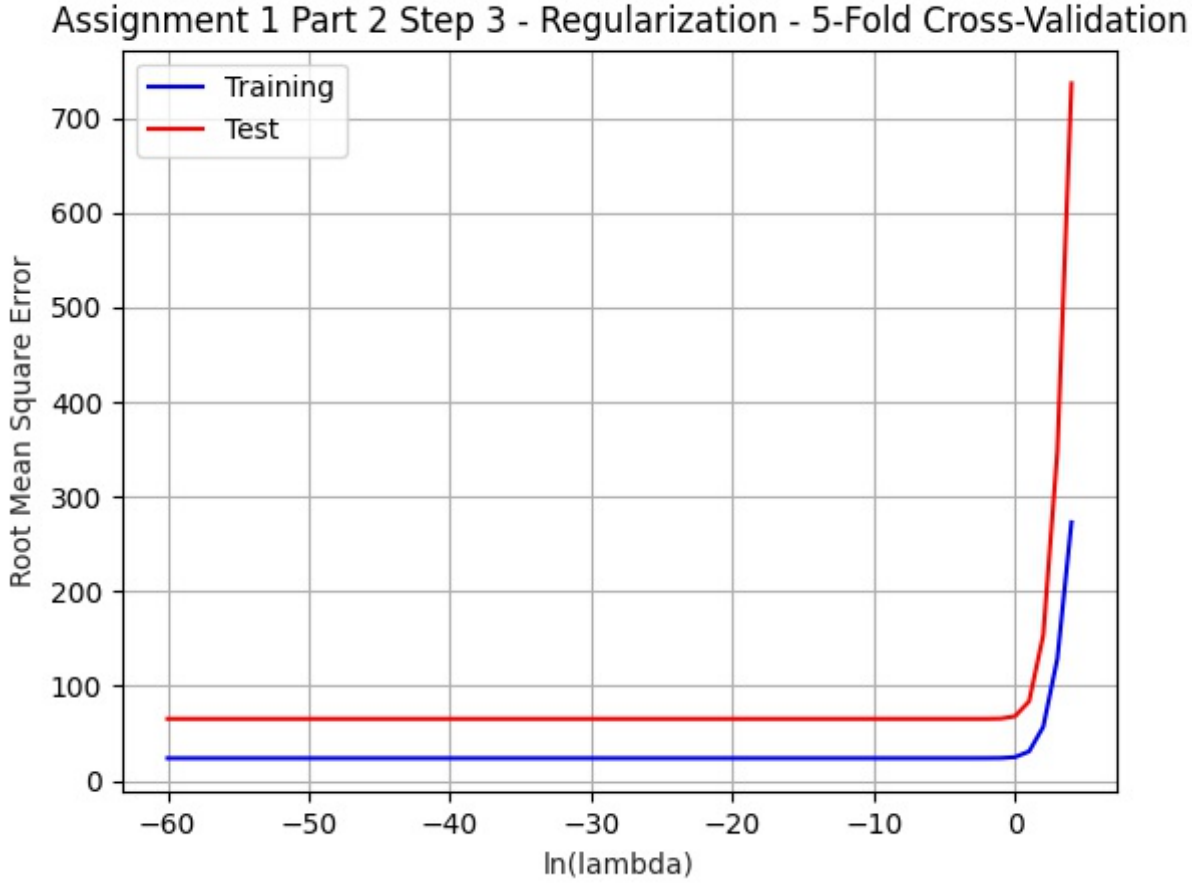

Assignment 1 Part 2 Step 2 - Predictions - No Cross-Validation
Root Mean Square Error
Predicted y
Target
Target Value
Sample Number
2500
2000
1500
1000
500
0
0
200
400
600
800
1000


Assignment 1 Part 2 Step 2 - Regularization - 5-Fold Cross-Validation
Training
Test
Root Mean Square Error
ln(lambda)
700
600
500
400
300
200
100
0
−60
−50
−40
−30
−20
−10
0

Step 3

```
Applying 5-fold cross validation
There were 500 independent variables and 1 dependent variables
There were 1000 samples in total
Completed in 52.595 milliseconds
Average Erms_train = 23.665797456855017
Average Erms_test = 65.28335649367837
```


Assignment 1 Part 2 Step 3 - Predictions - No Cross-Validation
Root Mean Square Error
Predicted y
Target
Target Value
Sample Number
2500
2000
1500
1000
500
0
0
200
400
600
800
1000


Assignment 1 Part 2 Step 3 - Regularization - 5-Fold Cross-Validation
Training
Test
Root Mean Square Error
ln(lambda)
700
600
500
400
300
200
100
0
−60
−50
−40
−30
−20
−10
0

Part 2 Overview

TODO