Rajesh Sakhamuru

6/13/2020

CS 6140 - Assignment # 2

2.1: Gradient Descent for Linear Regression

Housing Data:

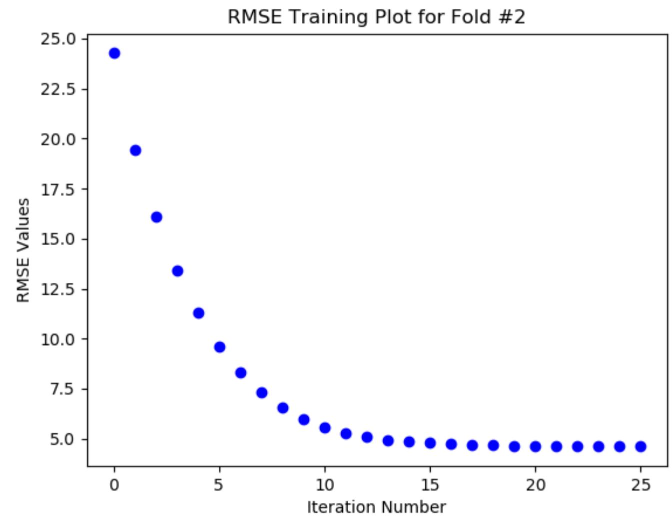
	Fold #	Train SSE	Train RMSE	Test SSE	Test RMSE
0	1	9968.563143	4.675562	1648.580287	5.685517
1	2	9692.914584	4.610465	2121.960398	6.450354
2	3	10604.737575	4.822448	895.040299	4.189249
3	4	10613.832752	4.824515	922.459235	4.252933
4	5	10898.738818	4.888839	641.665244	3.547065
5	6	10437.696303	4.784317	1084.063948	4.610440
6	7	9877.987279	4.654273	1759.615896	5.873864
7	8	10269.091178	4.745518	1338.631528	5.123249
8	9	10461.740051	4.789824	987.939753	4.401292
9	10	10542.678651	4.792577	951.424644	4.452117

Train data SSE Mean over folds: 10336.798033438758
Train data SSE Standard Deviation over folds: 360.2456707924791

Train data RMSE Mean over folds: 4.758833772534603
Train data RMSE Standard Deviation over folds: 0.08250341530395643

Test data SSE Mean over folds: 1235.1381232766305
Test data SSE Standard Deviation per fold: 444.2121612672998

Test data RMSE Mean over folds: 4.858607889792944
Test data RMSE Standard Deviation over folds: 0.8537681797169672



Yacht Data:

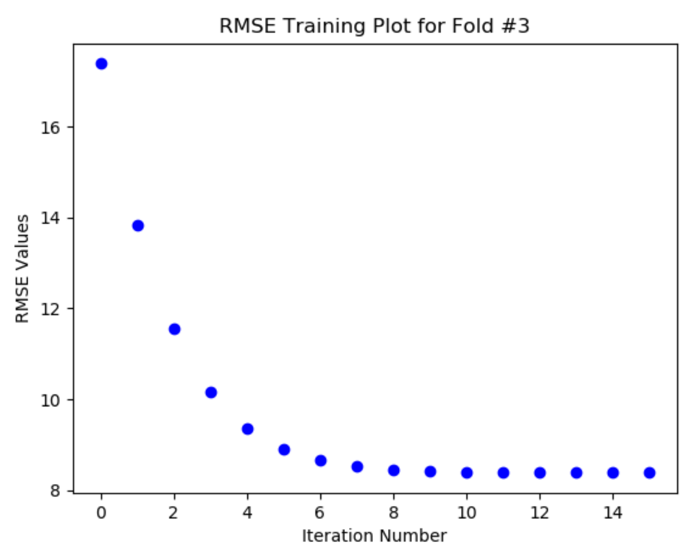
	Fold #	Train SSE	Train RMSE	Test SSE	Test RMSE
0	1	21469.008821	8.803718	2870.250317	9.622303
1	2	22559.288012	9.024493	1685.392777	7.373433
2	3	19452.806499	8.380140	4983.823073	12.679451
3	4	21615.391901	8.833680	2630.158245	9.211070
4	5	21527.564495	8.815715	2682.377029	9.302058
5	6	21828.616006	8.877143	2415.053415	8.826378
6	7	22448.582945	9.002323	1780.686351	7.579017
7	8	21831.410905	8.877711	2443.278256	8.877805
8	9	21697.791324	8.850501	2489.958761	8.962212
9	10	22370.318333	8.954348	1836.179344	7.957168

Train data SSE Mean over folds: 21680.077924121073
Train data SSE Standard Deviation over folds: 831.7483970141404

Train data RMSE Mean over folds: 8.84197727460605
Train data RMSE Standard Deviation over folds: 0.17022118869384428

Test data SSE Mean over folds: 2581.7157568042367
Test data SSE Standard Deviation per fold: 890.0606313434498

Test data RMSE Mean over folds: 9.03908967738749
Test data RMSE Standard Deviation over folds: 1.4087245953588408



Concrete Data:

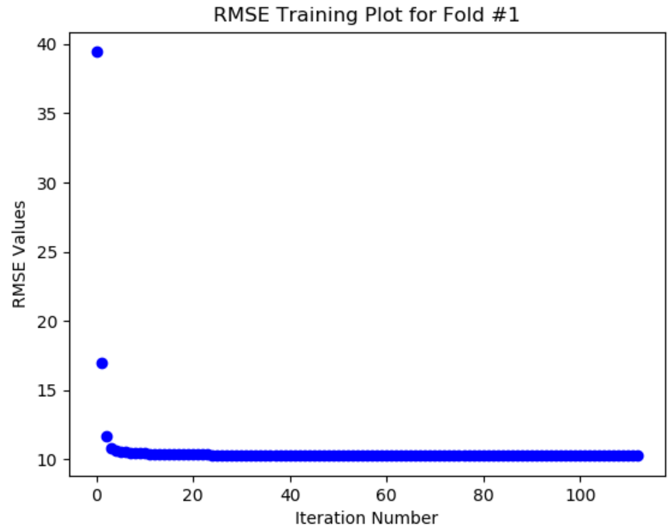
```
   Fold #      Train SSE  Train RMSE      Test SSE  Test RMSE
0       1   97111.419739   10.235175  13563.555580  11.475409
1       2   99486.391868   10.359576  11086.657080  10.374847
2       3   98161.787966   10.290379  12359.891945  10.954403
3       4   98543.396759   10.310362  12167.444883  10.868787
4       5  101369.257575   10.457148   9163.922628   9.432398
5       6  101118.334187   10.444198   9414.272006   9.560371
6       7   98208.179859   10.292810  12341.268714  10.946147
7       8  100228.950803   10.398165  10285.958980   9.993182
8       9  100873.125317   10.431527   9796.439483   9.752490
9      10   97934.946712   10.278482  12632.931166  11.074738

Train data SSE Mean over folds: 99303.5790785517
Train data SSE Standard Deviation over folds: 1437.837027436863

Train data RMSE Mean over folds: 10.349782183526294
Train data RMSE Standard Deviation over folds: 0.0749051403229818

Test data SSE Mean over folds: 11281.234246645665
Test data SSE Standard Deviation per fold: 1459.802217419942

Test data RMSE Mean over folds: 10.44327716630253
Test data RMSE Standard Deviation over folds: 0.6815483194335292
```



3: Least Squares Regression using Normal Equations

Housing Data:

Comparing Normal Equation results to Gradient Descent:

This is data on the right is comparing the first fold of the 10-fold cross validation, one using Normal Equations to calculate the weights, and the other using Gradient Descent. As we can see, the RMSE for both training and testing data is only minorly improved by using the Normal Equation and the corresponding weights are very similar. This indicates to me that given enough iterations of Gradient Descent, the weights and RMSEs will approach, but may not reach, the weights and RMSEs of the Normal Equations. This pattern persists through all 10 folds, where the average train and test RMSE over 10 folds for Normal Equations is just slightly lower than for Gradient Descent.

```
Weights using Normal Equations:
[[22.63118102]
 [-1.07717059]
 [ 1.12067134]
 [ 0.04632201]
 [ 0.67275866]
 [-1.94811184]
 [ 2.77057184]
 [-0.14887108]
 [-3.15812557]
 [ 2.68606407]
 [-2.05883418]
 [-2.0947837 ]
 [ 0.79545757]
 [-3.56260497]]
Training SSE: 9324.297755950205
Training RMSE: 4.52194884918302
Testing SSE: 1822.40016895513
Testing RMSE: 5.977736749487971
------------------
Weights for normalized data using Gradient Descent:
[[22.53280989]
 [-0.80163686]
 [ 0.62307637]
 [-0.41526375]
 [ 0.75728635]
 [-0.99648829]
 [ 3.25825463]
 [-0.29624198]
 [-2.18485004]
 [ 0.97996429]
 [-0.56833437]
 [-1.87057759]
 [ 0.8272415 ]
 [-3.27530159]]
Training SSE: 9684.794292193466
Training RMSE: 4.608533820436656
Testing SSE: 1884.5230968438573
Testing RMSE: 6.078769058900674
```

Yacht Data:

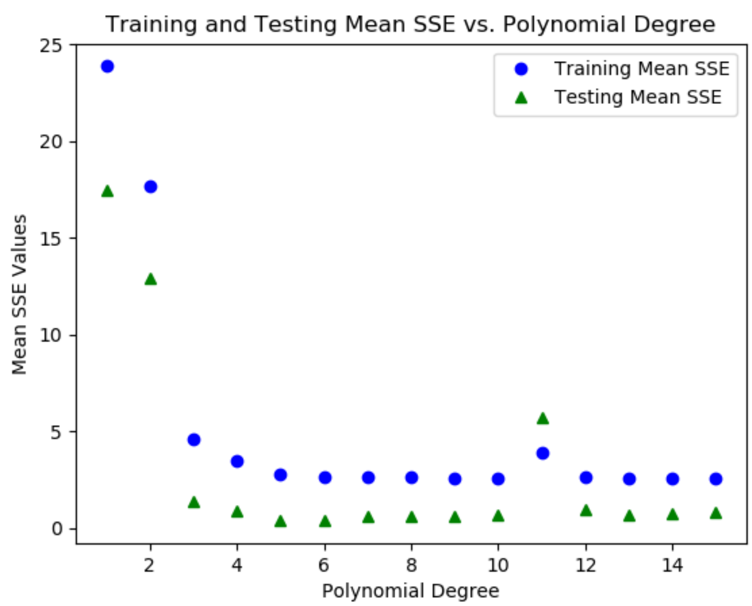
This is data on the right is comparing the first fold of the 10-fold cross validation, one using Normal Equations to calculate the weights, and the other using Gradient Descent. Very similarly to the Housing data Normal Equation results, the RMSE values are slightly improved compared to when using Gradient Descent. The weights are also similar, although the some weights are very different (like w_3 and w_4) while the RMSEs are still comparable; this could indicate a feature which has little bearing on the target prediction. The pattern of w_3 and w_4 being significantly different between using Normal Equations vs Gradient descent persists throughout the 10-folds.

```
------------------
Weights using Normal Equations:
[[10.40931331]
 [-0.0274754 ]
 [-0.27344196]
 [ 0.98151392]
 [-1.00423309]
 [-1.30546709]
 [12.09823922]]
Training SSE: 21590.534996502898
Training RMSE: 8.828599530076406
Testing SSE: 2644.8462579041643
Testing RMSE: 9.23675382494927
------------------
Weights for normalized data using Gradient Descent:
[[10.32886419]
 [-0.02920245]
 [-0.54596422]
 [-0.11225574]
 [-0.08640988]
 [-0.23014175]
 [12.00059425]]
Training SSE: 21601.62175923975
Training RMSE: 8.830865986791885
Testing SSE: 2657.4929400384985
Testing RMSE: 9.258810869162373
--------------------------------------------------
```

5.1: Polynomial Regression using Normal Equations

Sinusoid Data:

Polynomial Degree	Train Mean SSE	Train RMSE	Test Mean SSE	Test RMSE
1	23.861100	4.884782	17.490095	4.182116
2	17.639727	4.199967	12.925608	3.595220
3	4.571774	2.138171	1.394250	1.180784
4	3.481763	1.865948	0.876392	0.936158
5	2.751402	1.658735	0.369769	0.608086
6	2.654626	1.629302	0.408871	0.639430
7	2.602123	1.613110	0.639689	0.799805
8	2.602072	1.613094	0.641988	0.801241
9	2.589897	1.609316	0.616145	0.784949
10	2.559947	1.599983	0.711988	0.843794
11	3.882091	1.970302	5.678733	2.383009
12	2.664466	1.632319	0.963790	0.981728
13	2.588912	1.609010	0.658033	0.811192
14	2.578215	1.605682	0.762671	0.873311
15	2.587556	1.608588	0.790020	0.888831



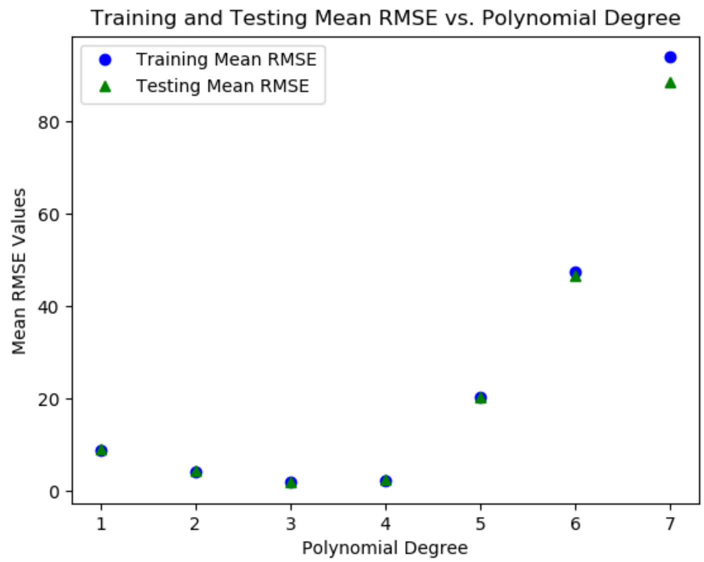
As we can see from this data, the best fit for the data is when using a Polynomial Degree of 5 because it has the lowest Testing Mean SSE.

Because there is only one feature, and because it is a Sin graph, it is not massively impacted by an increase in Polynomial Degree, unless you get to polynomial degree >20 which is generally not used anyhow.

5.1 cont…

Yacht Data:

Polynomial Degree	Train RMSE	Test RMSE
1	8.841490	9.011309
2	4.071190	4.268255
3	1.813776	1.939529
4	2.233009	2.377268
5	20.293248	20.228107
6	47.464602	46.691472
7	93.952099	88.502518



We can clearly see from this data that the best polynomial degree for this data is polynomial degree 3. The Train and Test RMSE is the lowest, and once the degree is >4, the error increases rapidly.

7: Programming Ridge Regression

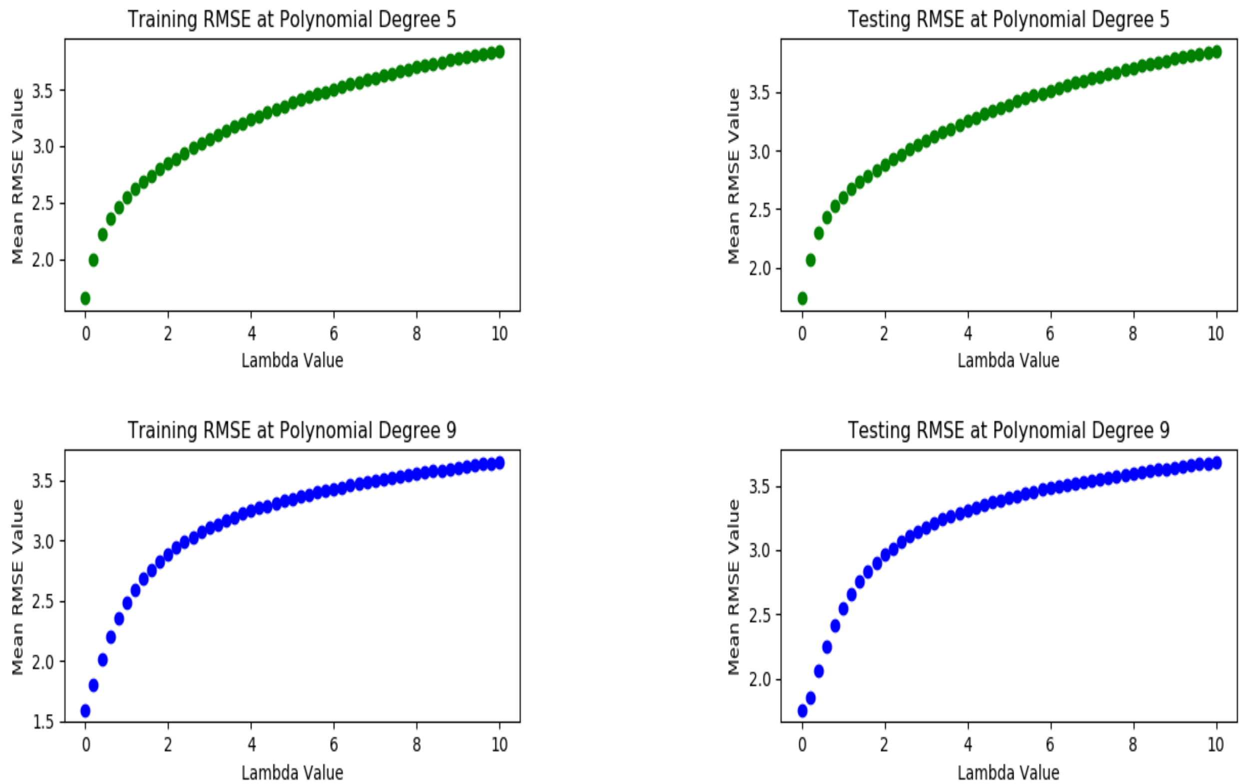
Data for Polynomial degree 5 and 9 are below, and the graphs for both are immediately following on the next page.

Polynomial Degree: 5

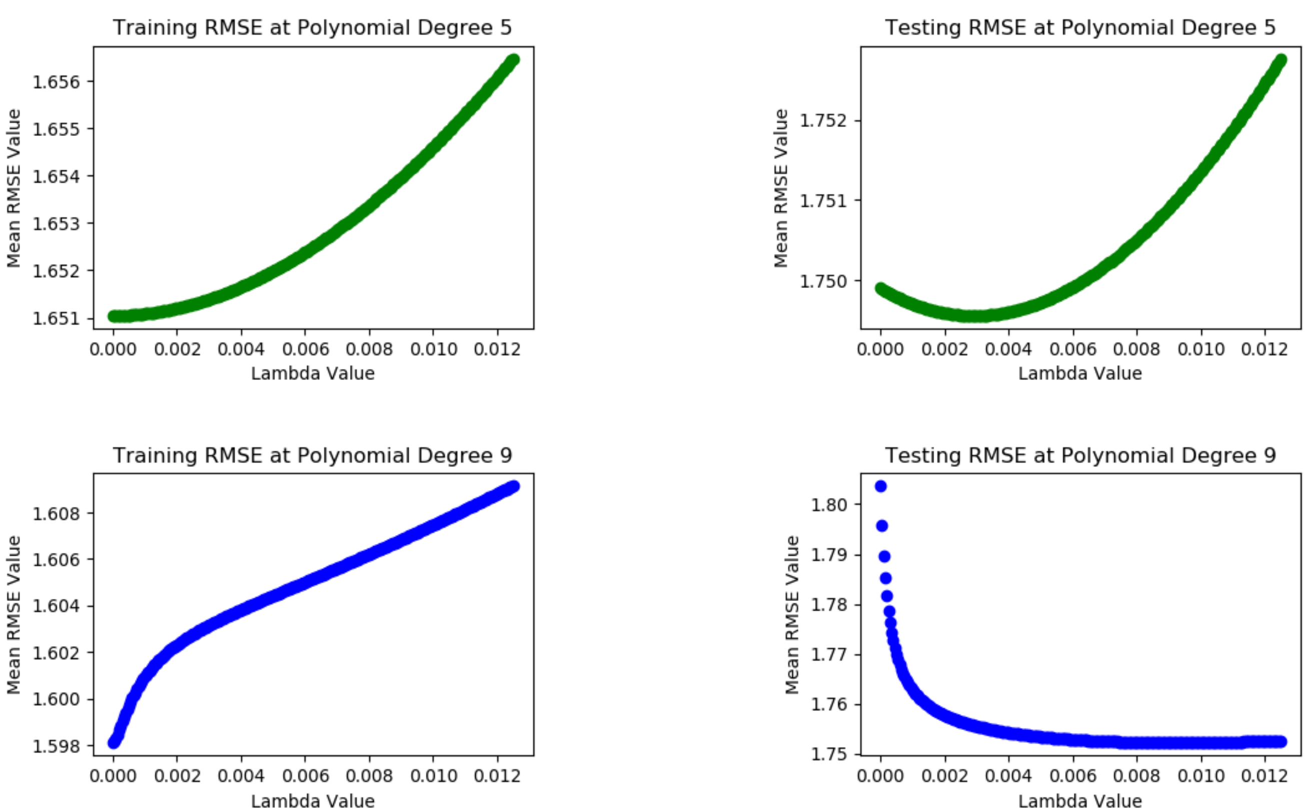
	Lambda Values	Train RMSE	Test RMSE
0	0.0	1.651230	1.738041
1	0.2	1.990620	2.066267
2	0.4	2.220797	2.294594
3	0.6	2.362604	2.432205
4	0.8	2.465250	2.529510
5	1.0	2.547827	2.606486
6	1.2	2.618714	2.671937
7	1.4	2.682030	2.730173
8	1.6	2.739994	2.783489
9	1.8	2.793890	2.833185
10	2.0	2.844511	2.880042
11	2.2	2.892376	2.924549
12	2.4	2.937844	2.967034
13	2.6	2.981178	3.007722
14	2.8	3.022577	3.046777
15	3.0	3.062199	3.084327
16	3.2	3.100177	3.120472
17	3.4	3.136622	3.155297
18	3.6	3.171631	3.188875
19	3.8	3.205291	3.221270
20	4.0	3.237678	3.252542
21	4.2	3.268864	3.282742
22	4.4	3.298913	3.311921
23	4.6	3.327885	3.340125
24	4.8	3.355834	3.367399
25	5.0	3.382813	3.393783
26	5.2	3.408870	3.419316
27	5.4	3.434050	3.444035
28	5.6	3.458393	3.467975
29	5.8	3.481942	3.491170
30	6.0	3.504731	3.513650
31	6.2	3.526797	3.535447
32	6.4	3.548171	3.556589
33	6.6	3.568886	3.577102
34	6.8	3.588971	3.597012
35	7.0	3.608452	3.616345
36	7.2	3.627357	3.635123
37	7.4	3.645710	3.653369
38	7.6	3.663533	3.671104
39	7.8	3.680850	3.688348
40	8.0	3.697681	3.705119
41	8.2	3.714047	3.721438
42	8.4	3.729965	3.737319
43	8.6	3.745453	3.752781
44	8.8	3.760529	3.767840
45	9.0	3.775209	3.782509
46	9.2	3.789508	3.796804
47	9.4	3.803440	3.810737
48	9.6	3.817019	3.824323
49	9.8	3.830258	3.837574
50	10.0	3.843171	3.850502

Polynomial Degree: 9

	Lambda Values	Train RMSE	Test RMSE
0	0.0	1.598622	1.760834
1	0.2	1.799556	1.850490
2	0.4	2.014822	2.061146
3	0.6	2.201663	2.254777
4	0.8	2.356686	2.416621
5	1.0	2.485093	2.550232
6	1.2	2.592499	2.661315
7	1.4	2.683478	2.754766
8	1.6	2.761522	2.834373
9	1.8	2.829267	2.902998
10	2.0	2.888711	2.962808
11	2.2	2.941384	3.015455
12	2.4	2.988471	3.062217
13	2.6	3.030899	3.104092
14	2.8	3.069406	3.141868
15	3.0	3.104583	3.176177
16	3.2	3.136909	3.207528
17	3.4	3.166774	3.236339
18	3.6	3.194504	3.262951
19	3.8	3.220365	3.287649
20	4.0	3.244586	3.310671
21	4.2	3.267354	3.332217
22	4.4	3.288834	3.352457
23	4.6	3.309162	3.371535
24	4.8	3.328457	3.389577
25	5.0	3.346823	3.406689
26	5.2	3.364349	3.422963
27	5.4	3.381112	3.438481
28	5.6	3.397181	3.453314
29	5.8	3.412616	3.467522
30	6.0	3.427468	3.481161
31	6.2	3.441787	3.494279
32	6.4	3.455612	3.506918
33	6.6	3.468982	3.519117
34	6.8	3.481929	3.530910
35	7.0	3.494483	3.542326
36	7.2	3.506673	3.553393
37	7.4	3.518520	3.564137
38	7.6	3.530049	3.574578
39	7.8	3.541277	3.584737
40	8.0	3.552225	3.594632
41	8.2	3.562907	3.604280
42	8.4	3.573340	3.613695
43	8.6	3.583536	3.622891
44	8.8	3.593509	3.631882
45	9.0	3.603270	3.640677
46	9.2	3.612830	3.649288
47	9.4	3.622199	3.657724
48	9.6	3.631385	3.665995
49	9.8	3.640398	3.674108
50	10.0	3.649244	3.682070



Graph Below is the same model implementation but using lambda values between 0 and 0.0125. It is referenced in **7.1** Interpretation below:



7.1: Interpretation

Graphing the test and train RMSE at the values of lambda (between 0 and 10 at .2 intervals) given by the exercise looks like at all values of lambda, the RMSE increases, leading to worse predictions using test data. This is very unexpected as we would expect the RMSE to decrease before increasing and that there would be an optimal value of lambda which improves the testing RMSE.

Because of this unusual result, I ran the model again, but this time using lambda values between 0 and 0.0125 at both polynomial degree 5 and 9. The result in this case was far more convincing. At polynomial degree 5, the mean RMSE value for test data decreases until lambda ≈ 0.003 (by less than 0.001 RMSE), before increasing again. Similarly, at polynomial degree 9, the mean RMSE value for test data decreases until lambda ≈ 0.0095 before increasing again.

For both synthetic datasets, we can see that as the value of lambda increases, the model complexity decreases and reduces overfitting, but once the optimal lambda threshold is crossed, it can cause significant underfitting which can be seen in this case due to increasing RMSE and leveling off at around a limit value of 4. which is essentially making random predictions unrelated to the training data.

Comparing the two synthetic datasets, we can see that the optimal lambda value at the higher, less-optimal polynomial degree of 9 is impacted by reducing the RMSE by about 0.5 . The polynomial degree of 5, is much less impacted and at the optimal lambda the RMSE is only decreased by about 0.0005. Thus we can make an obvious conclusion that the polynomial degree used to fit the data has far more of an impact on the RMSE than the lambda value.

Another benefit of using ridge regression, is that, if ANY value of lambda is used (although an unnecessarily high value will still be very underfit) the RMSE is limited regardless of polynomial degree, even absurdly high polynomial degrees like 100. This demonstrates the ability of ridge regression to prevent overfitting in a model versus just the linear regression (aka lambda = 0).