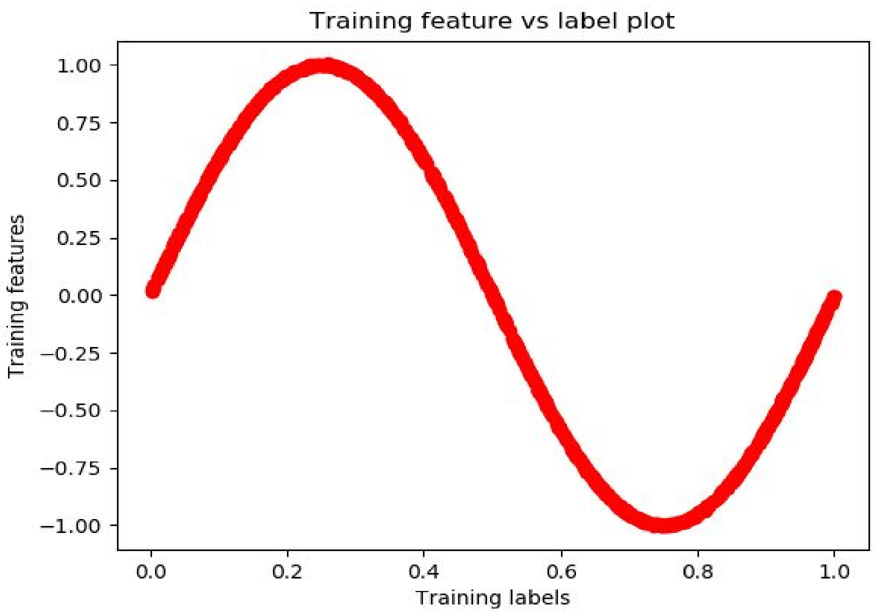
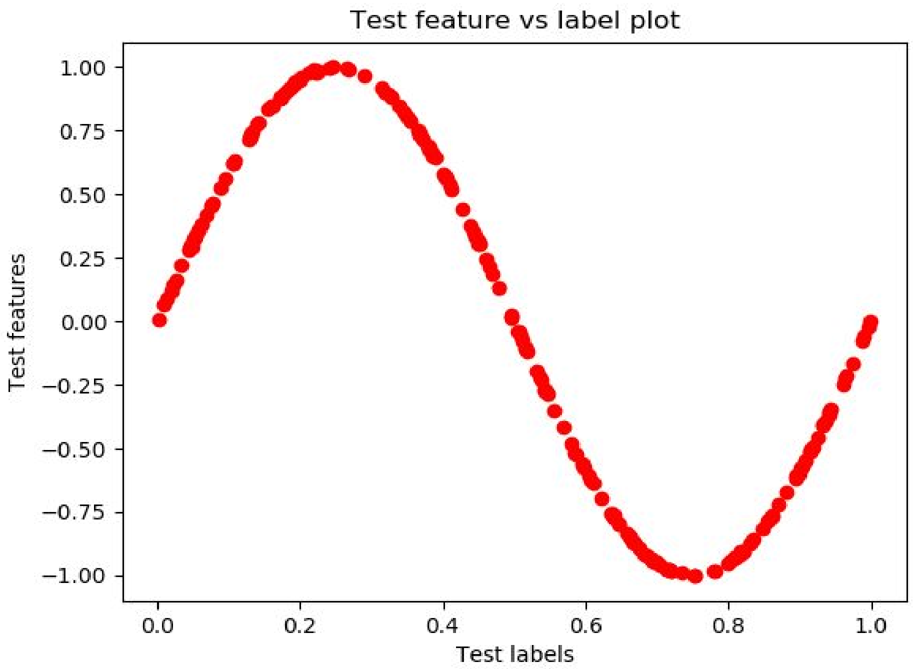
Name: Gurjot Singh Suri
Roll No. 17CS10058

Tried many convergence criteria like fixing the number of iterations and fixing the difference of squared errors. I have used a difference of 10^{-8} between consecutive losses as the criteria for all the parts as it could fit the training data in an acceptable amount of time. Fixing the number of iterations equal to $5*10^6$ gave better results but took a lot more time to run.

1)a)





1)b)

Degree 1:
Training Error: 0.09968054237094173
Test Error: 0.09553057205696289

Polynomial Term	Co-efficient learnt
1	0.91609634
x	-1.85516555

Degree 2:
Training Error: 0.09914021385068623
Test Error: 0.09579854742921143

Polynomial Term	Co-efficient learnt
1	0.97374768
x	-2.2009799
x^2	0.34054149

Degree 3:
Training Error: 0.003239249117089172
Test Error: 0.003248849012703992

Polynomial Term	Co-efficient learnt
1	-0.07671109
x	10.50647865
x^2	-31.22342863
x^3	20.91010109

Degree 4:
Training Error: 0.004617705375705581
Test Error: 0.004675270139331303

Polynomial Term	Co-efficient learnt
1	0.08300767
x	7.17897137
x^2	-15.64847551
x^3	-3.94947702
x^4	12.65368186

Degree 5:
Training Error: 0.008654167959586343
Test Error: 0.008861655970749617

Polynomial Term	Co-efficient learnt
1	0.19217578
x	5.40864809
x^2	-10.24494945
x^3	-4.97037268
x^4	2.52120717
x^5	7.55024148

Degree 6:
Training Error: 0.004544300590550844
Test Error: 0.004590889734806446

Polynomial Term	Co-efficient learnt
1	0.0718704
x	7.23242959
x^2	-15.80248417
x^3	-2.21222327
x^4	7.14273297
x^5	6.20060081
x^6	-2.30561793

Degree 7:
Training Error: 0.002337724529694965
Test Error: 0.0023336651000341103

Polynomial Term	Co-efficient learnt
1	0.03406228
x	7.64731231
x^2	-16.09754347
x^3	-3.7202358
x^4	6.58935788
x^5	8.27073487
x^6	3.22071671
x^7	-5.74997768

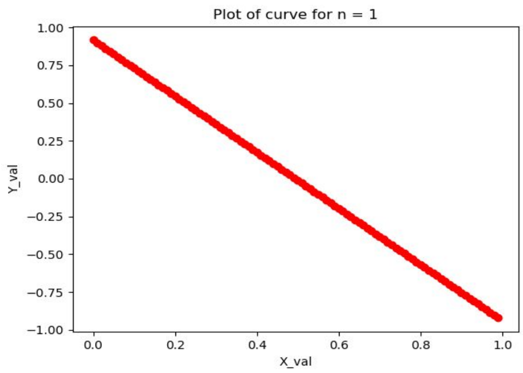
Degree 8:
Training Error: 0.001432137497172644
Test Error: 0.0014136227171906394

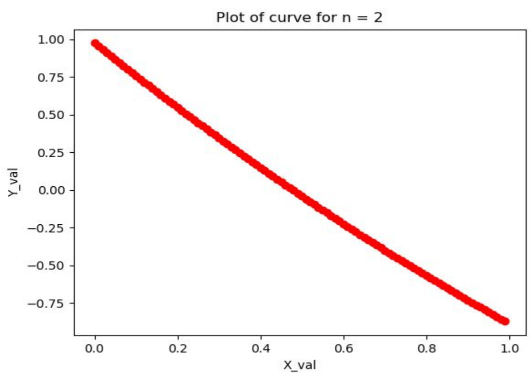
Polynomial Term	Co-efficient learnt
1	0.03685663
x	7.44790523
x^2	-14.81220748
x^3	-4.95664486
x^4	4.65089426
x^5	7.86059968
x^6	5.78443294
x^7	0.48688848
x^8	-6.4101526

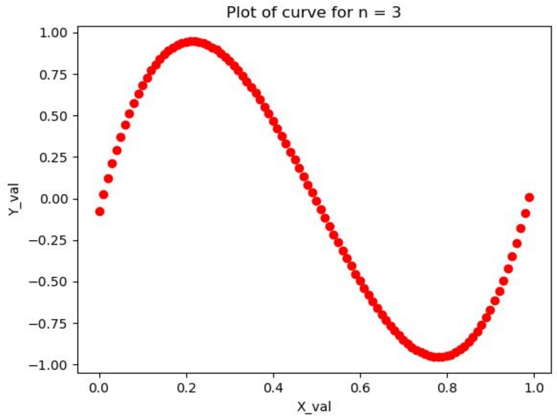
Degree 9:
Training Error: 0.0012223746686057298
Test Error: 0.0012155775122382672

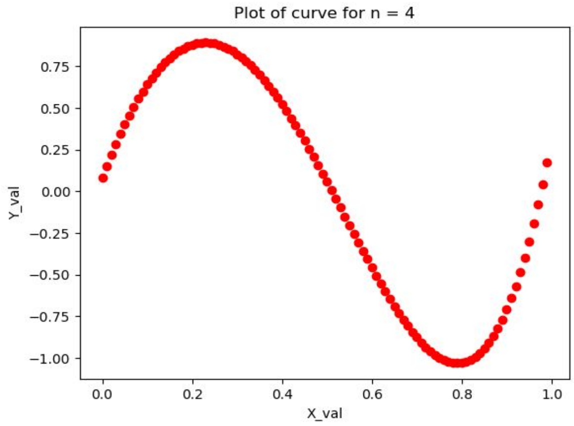
Polynomial Term	Co-efficient learnt
1	0.05620563
x	7.05636377
x^2	-13.33257186
x^3	-5.56780733
x^4	2.94016501
x^5	6.6608546
x^6	6.2376834
x^7	3.2114854
x^8	-1.17215435
x^9	-6.09006228

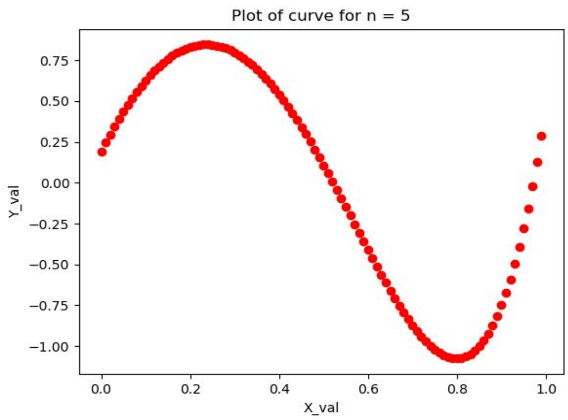
2) a)

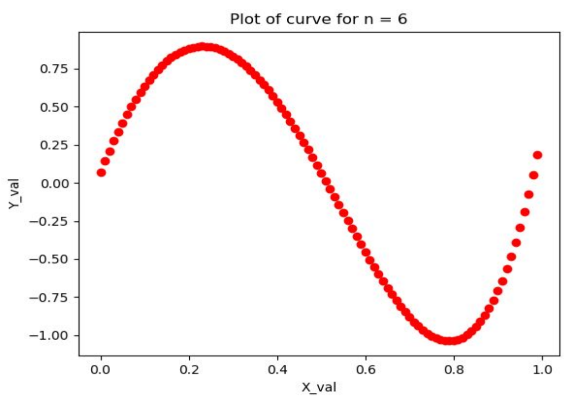


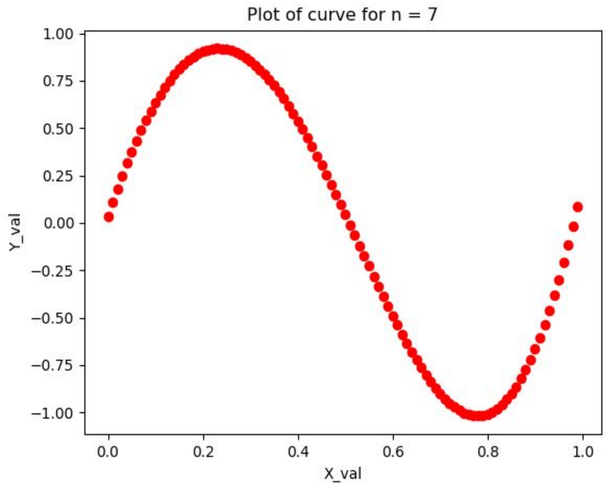


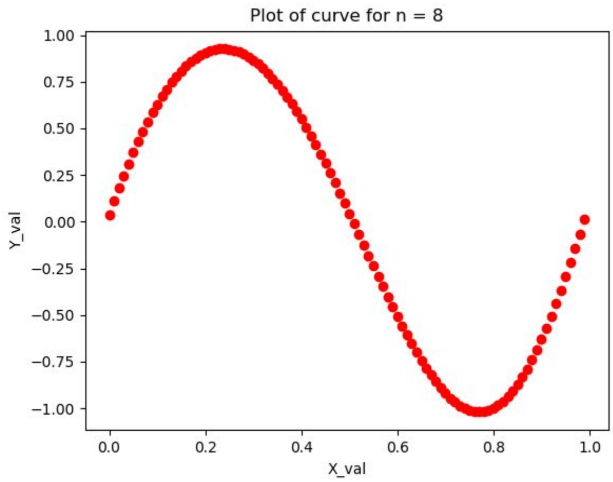

Plot of curve for n = 3
Y_val
X_val
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0

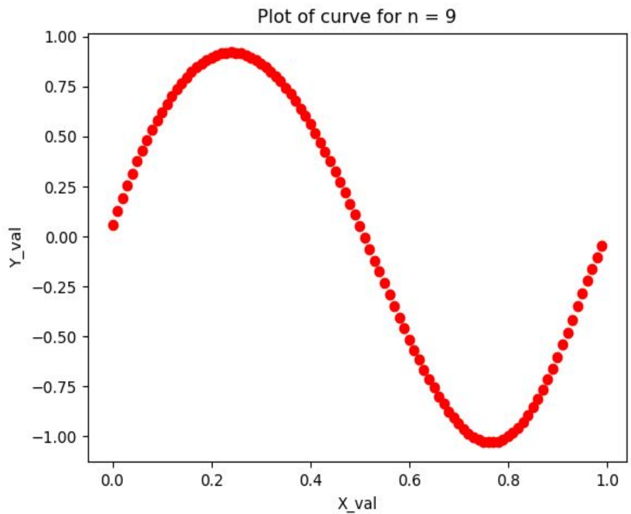

Plot of curve for n = 4
Y_val
X_val
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0


Plot of curve for n = 5
Y_val
X_val
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0


Plot of curve for n = 6
Y_val
X_val
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0

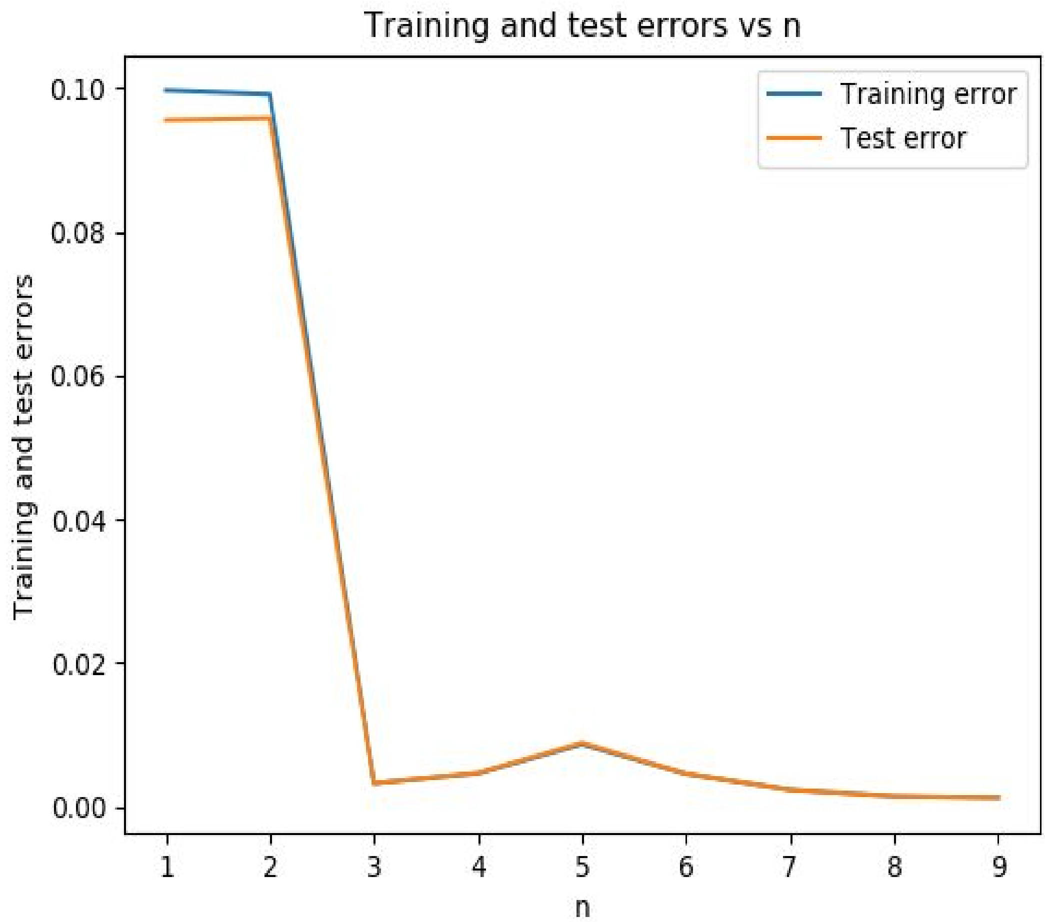






Plot for n = 9 seems most similar to the training dataset while n = 1 seems the most different from the original curve.

2) b)



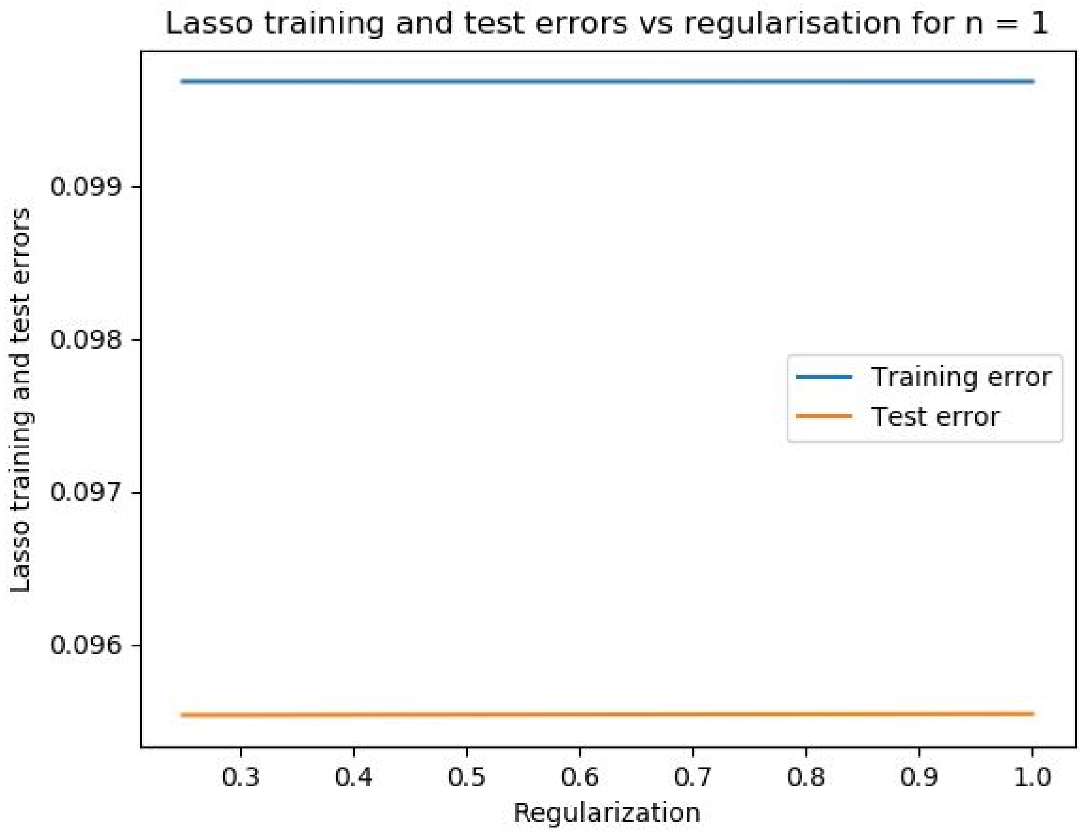
Explain which value of n is suitable for the dataset that you have, and why.
Both the training and test losses versus n generally decrease with increase in n as shown in the graph except for n = 3, where the loss is a little lower.

N=9 is most suitable for this dataset as both the training and test losses are the lowest for this value of n. Also the curve for n=9 in 2) a) matches the training dataset most closely. The dataset seems to be a sine function whose expansion consists of polynomial degrees, for n=9, we are able to approximate it for the most number of terms.

3) a)

Lasso, N = 1 (maximum training error in part 2)

Regularization parameter	Training error	Test error
0.25	0.09968022792486098	0.09553389924245824
0.5	0.09967999242260117	0.09553672808685418
0.75	0.09967980597872034	0.0955392978967117
1.0	0.09967967915391852	0.09554127915980505



Training errors decrease slightly with increase in regularization parameter.
Test errors increase slightly with increase in regularization parameter.

Lasso, N = 9 (minimum training error in part 2)

Regularization parameter	Training error	Test error
0.25	0.0012220960557283696	0.0012179143156387038
0.5	0.0012224808269620492	0.0012208992096101122
0.75	0.0012235489699932178	0.0012245526211911329
1.0	0.0012252904728498849	0.0012288643103671754

Both training and test errors increase significantly with increase in regularization parameter.

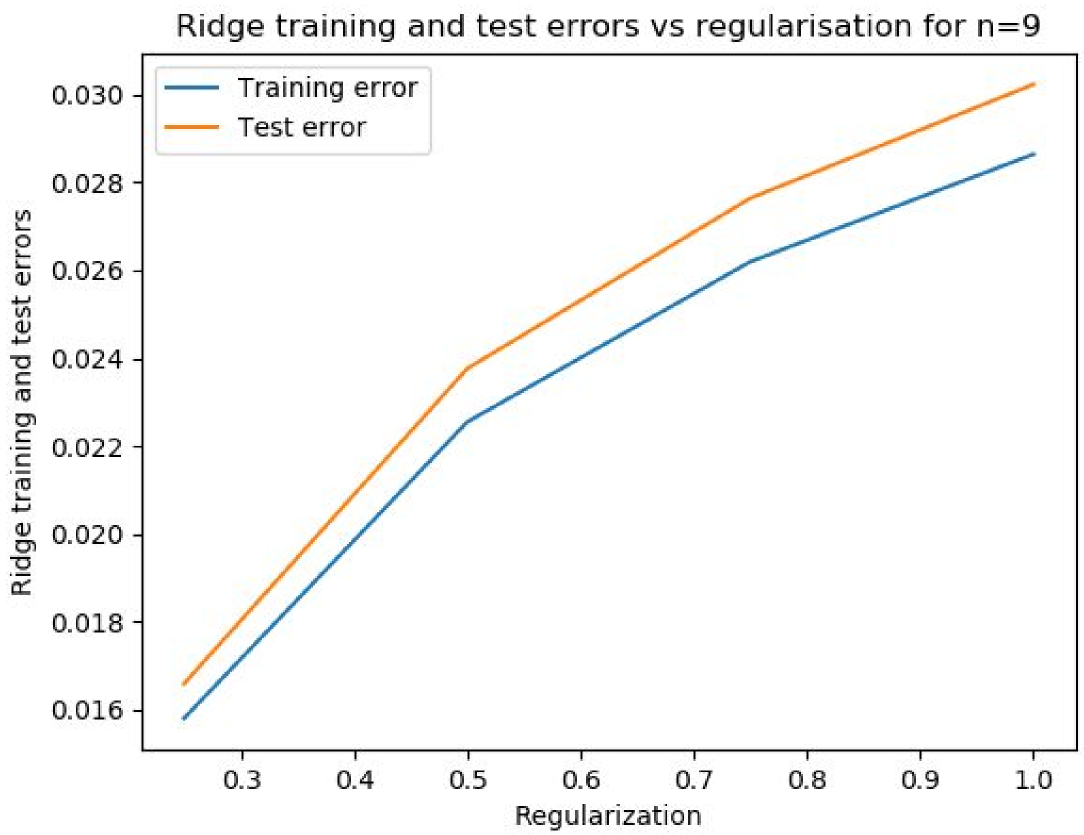
3) b)
Ridge, N = 1 (maximum training error in part 2)

Regularization parameter	Training error	Test error
0.25	0.09968259809327043	0.09551486266666791
0.5	0.0996865113599861	0.09549576643865666
0.75	0.09969259155335454	0.09547590473063139
1.0	0.09970092758275782	0.09545680313913259

Training errors increase slightly with increase in regularization parameter.
Test errors decrease slightly with increase in regularization parameter.

Ridge, N = 9 (minimum training error in part 2)

Regularization parameter	Training error	Test error
0.25	0.015805872212083155	0.016592673178909195
0.5	0.022549258184624242	0.023763758392800097
0.75	0.026196697665996867	0.02764190172575355
1.0	0.028642570549437908	0.03023322943038642



Both training and test errors increase significantly with increase in regularization parameter.

What differences do you notice between the two kinds of regression?
In case of Lasso regression, the training loss for n =1 and lamda=1 is less than the training loss for normal regression of part 2 for n=1. Also the training loss for n =9 and lamda=0.25 is less than the training loss for normal regression of part 2 for n=9. For n=9, the losses are close to and for some values of lamda are better than normal regression losses. In case of n=1, training errors decrease slightly with increase in lamda while for other cases of lamda and n, it increases.

On the other hand for Ridge regression, the training and test losses both are consistently higher than that of normal regression and lasso regression for corresponding values of n and lamda. In case of n=1, test errors decrease slightly with increase in lamda while for other cases, it increases.

Which one would you prefer for this problem and why?
I would prefer lasso regression with n=9 and lamda=0.25 as it gives the best training losses as compared to other combinations of regression, n and lamda. Also, the training and test losses both are consistently lower than that of ridge regression for corresponding values of n and lamda.

For lasso regression and n=9, the best training loss is 0.0012220960557283696
For ridge regression and n=9, the best training loss is 0.015805872212083155 which is a lot higher.

This may be because Lasso regression overcomes the disadvantage of Ridge regression by not only punishing high values of the weights but actually setting them to zero if they are not relevant. Therefore, we might end up with fewer features included in the model than we started with, which is a huge advantage.

But even in the lasso case there is not much improvement than the normal regression and could be avoided.