MACHINE LEARNING

ASSIGNMENT 1: LINEAR REGRESSION

SARTHAK CHAKRABORTY (16CS30044)

1. Synthetic Data Generation and Simple Curve Fitting

a) Synthetic Data has been generated as instructed in the question. The input values $\{x_i\}$ are generated uniformly in the range[0, 1], and the corresponding target values $\{y_i\}$ are obtained by first computing the corresponding values of the function sin(2Πx), and then adding a random noise with a Gaussian distribution having standard deviation 0.3. 10 instances of (x_i, y_i) are generated. The dataset is saved in the file '**dataset.csv**'. An instance of the dataset is given in Fig. 1.

X	Y
0.905105	-0.42824
0.329617	1.464201
0.87819	-1.26109
0.389926	0.698419
0.296615	1.640573
0.962315	-0.11728
0.619805	-0.62656
0.617476	-0.2428
0.694025	-1.11403
0.329307	1.118865

Fig. 1: An instance of the data

b) The dataset is then split into the training set (80%) and test set (20%)

c) A polynomial of degree n is fit to the dataset with *Squared Error Cost Function* using gradient descent. The value of n is varied from 1 to 9. The weights of the fit generated are then used to measure error on the training as well as the test set. The *learning rate* used for gradient descent is *0.05*.

[**N.B.: The stopping condition for the gradient descent algorithm is maintained by limiting the number of iterations. I have used 5000 iterations of the algorithm in Part 1 of the question. The weights are initialized randomly from a uniform distribution U(0,1) before the gradient descent algorithm.**]

For the data generated as shown in Fig. 1, the learned weights of the fits are shown as below:

Degree of Polynomial n: 1
Weights Learned:
[[2.51247577 -4.22188682]]
Training Error: 0.10279052800437628
Test Error: 1.026792663890925

Degree of Polynomial n: 2
Weights Learned:
[[2.13701953 -2.94057532 -0.94721865]]
Training Error: 0.11698705832624551
Test Error: 1.0550765136968814

Degree of Polynomial n: 3
Weights Learned:
[[2.49161477 -3.79764713 -2.00844087 1.94161267]]
Training Error: 0.08393589731441678
Test Error: 0.6906502268468534

Degree of Polynomial n: 4
Weights Learned:
[[2.80755035 -4.3502618 -2.58749464 0.84541515 2.42338489]]
Training Error: 0.052920797048179695
Test Error: 0.3411652179126553

Degree of Polynomial n: 5
Weights Learned:
[[2.86674745 -4.20771388 -2.66164124 -0.45717557 1.28611717 2.81523711]]
Training Error: 0.03776205567183047
Test Error: 0.13063023805355475

Degree of Polynomial n: 6
Weights Learned:
[[2.83685225 -3.76710654 -3.10926562 -1.04761297 1.07872243 1.43582463
2.71883324]]
Training Error: 0.029870751740051302
Test Error: 0.023063409339722386

Degree of Polynomial n: 7
Weights Learned:
[[2.81789323 -3.71124523 -2.56243701 -1.64102216 0.23646574 0.90727218
1.97104497 2.64463104]]
Training Error: 0.02599104279519679
Test Error: 0.002201199694041878

Degree of Polynomial n: 8
Weights Learned:
[[2.7073811 -3.10695437 -3.33408248 -1.22645865 -0.35105825 0.88594398
1.93206127 1.90335537 1.59355787]]
Training Error: 0.024446804544018992
Test Error: 0.03007416154319845

Degree of Polynomial n: 9
Weights Learned:
[[2.62665416 -2.84092014 -3.12951369 -1.766344 -0.61752244 1.10458576
1.68988484 1.15831467 1.3985287 1.81383657]]
Training Error: 0.023023882160387296
Test Error: 0.1054853941715343

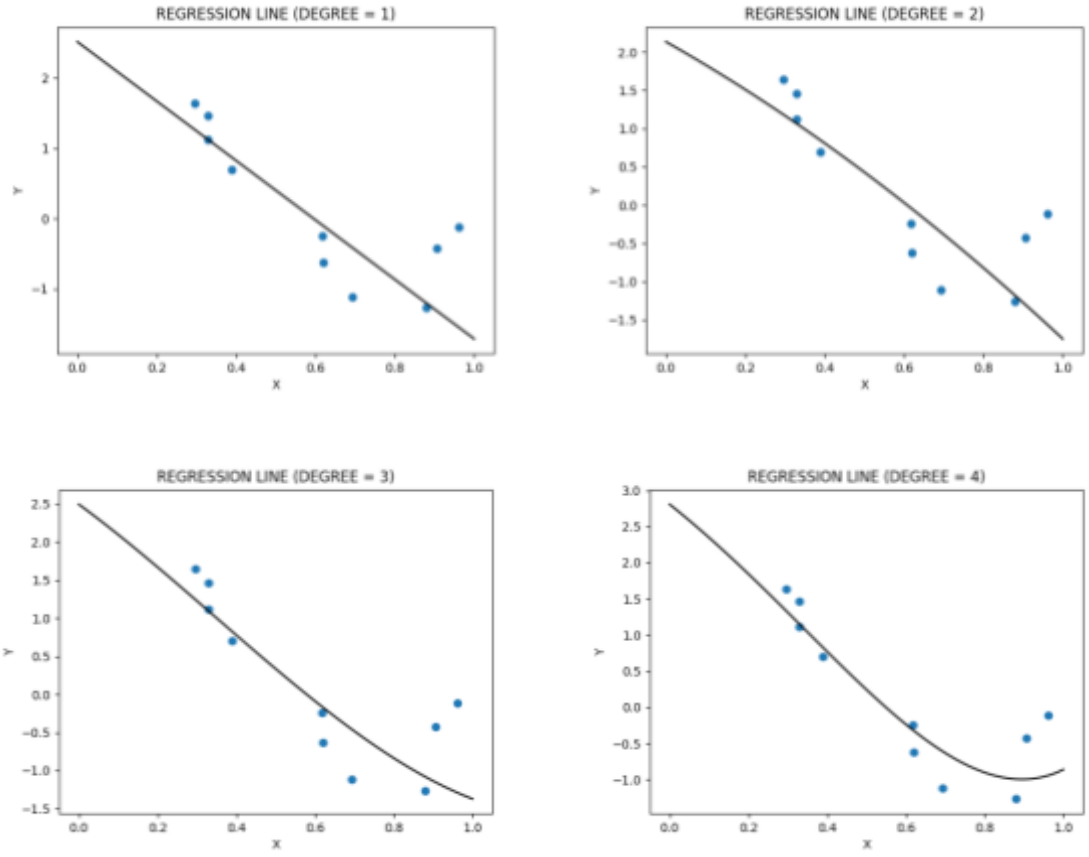
Thus, the values generated are shown. It can be seen that **minimum test error** occurs for the polynomial of **degree 7** for the given data and **minimum train error** occurs for the polynomial of **degree 9**. However it varies with dataset. No general comment can be made as such. The plots in the next section will show the fits graphically

[**N.B.: 'output.txt' file contains the results. The output currently in the file is not same as shown in the report since I have run the program a few more times and it has got updated.**]

---

2. Visualization of the dataset and the fitted curves

a) Made **separate plots** for all the 9 degrees of polynomials that is fitted on the dataset in Fig. 1. The plots are as below:



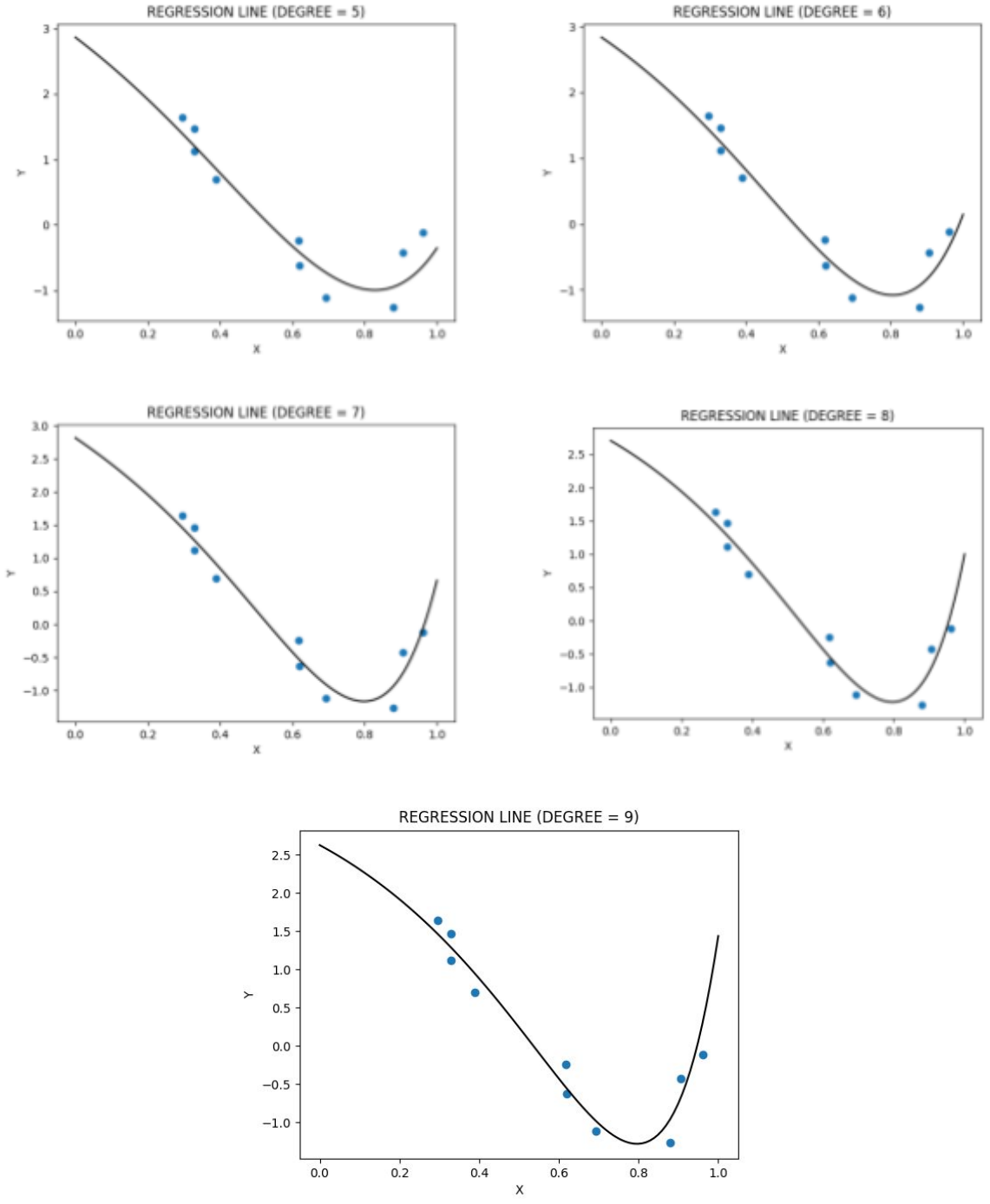


Fig. 2: Plots for the data fitted with curves of varying degree

The plots show the various curves fit for the dataset. We see that **the fit gets better as we increase the degree of the polynomial**. This is expected, since sin curve is approximated as an infinite degree polynomial by Taylor series. Hence a perfect sine curve will be best fit by a high degree polynomial.

b) Now, we will see the plot of Train Error and Test Error vs degree of polynomial for the given dataset.

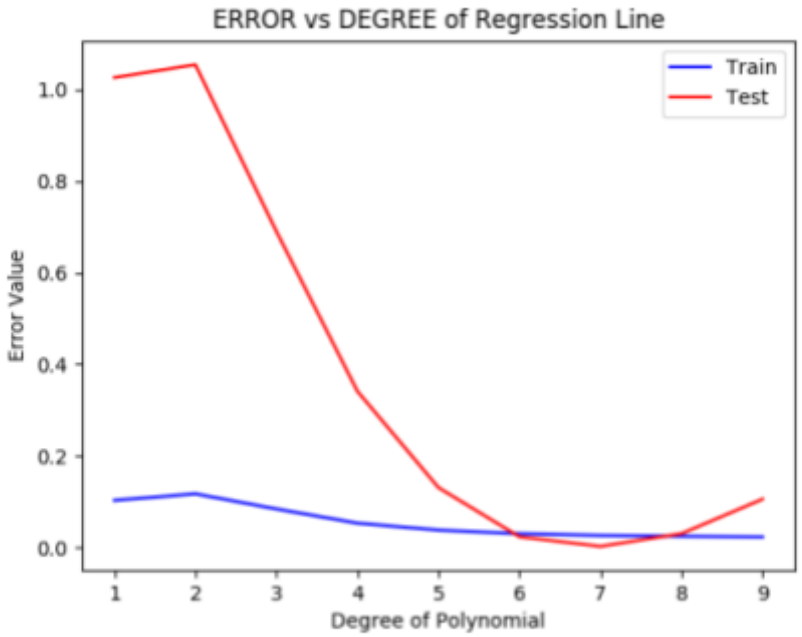


Fig. 3: Train Error and Test Error of the polynomial fitted in Fig. 2 on the dataset sown in Fig. 1 vs the degree of polynomial

In this plot, we see that the **train error decreases with the increase in degree of polynomial**. It has the lowest error for the curve having a **degree of 9**. However, the error on the **test set** first decreases rapidly with increase in degree of polynomial, achieves a **minimum at n=7** and then increases.

Thus, to the synthetic data that have been generated in Fig. 1, the most suitable value of n that will fit the data is **7 or 8**. This is because both the train and test error are simultaneously lowA curve with n=9 is giving higher error in test set. Hence, it is not suitable. However, this largely depends on the dataset and also the number of samples is very low (here, 10). A more appropriate prediction can be given with the increase in dataset.

Lets see, how the train error and test error vs degree of polynomial varies on an average for different dataset.

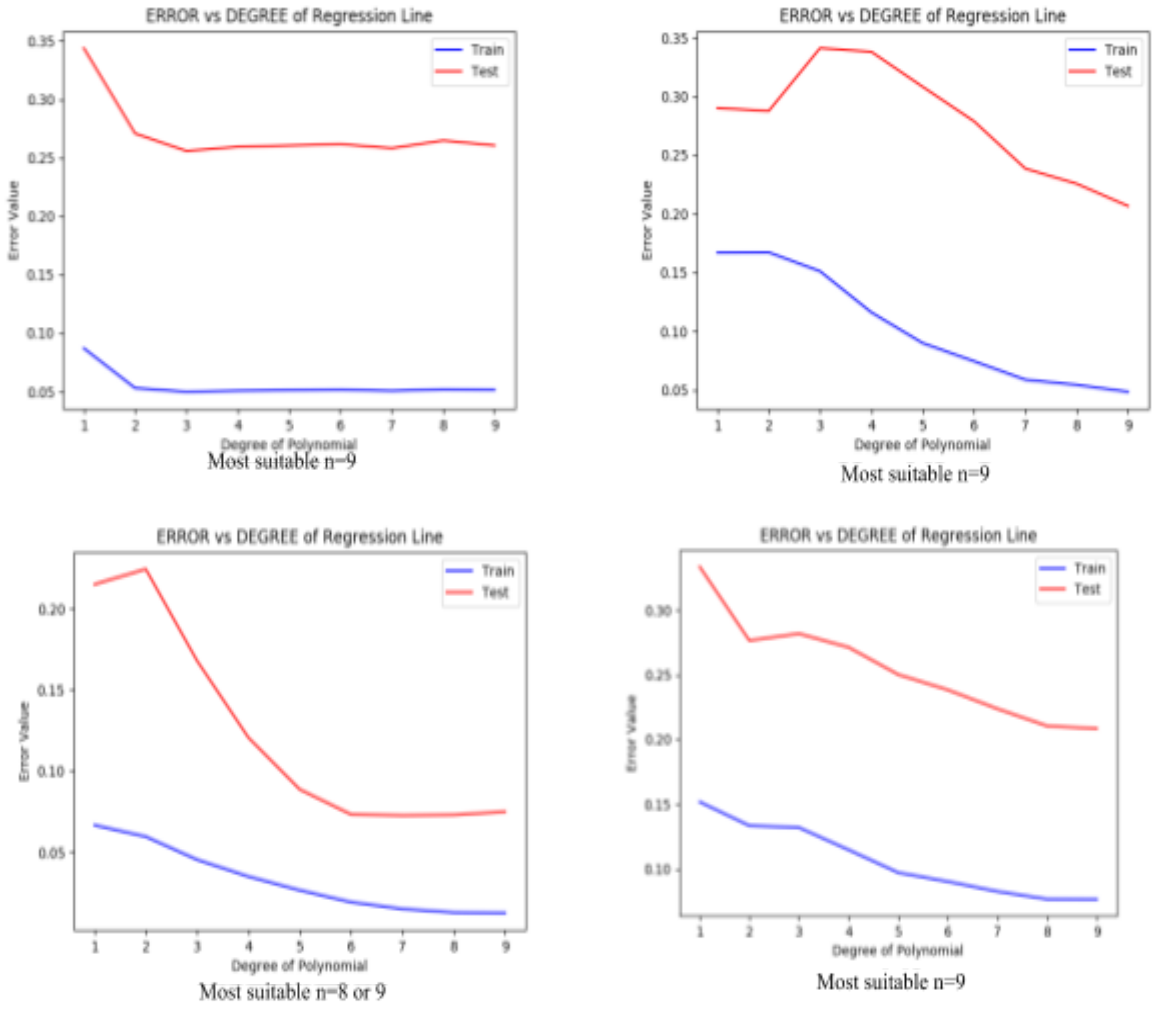


Fig. 4: Plot showing a general trend among Test error and Train error vs Degree of Polynomial for different datasets

Thus, we see that in general a higher degree of polynomial gives a less train and test error. So a **polynomial of degree 9** works for almost all datasets and fits the data nicely with low error.

3. Experimenting with a larger training set

a) Here I repeat the earlier part 1 and 2 with different sizes of the dataset. Datasets of size **M=100, 1000, and 10000** are used to get the parameter values. I have plotted the fitted curves and the train and test error vs. degree of polynomial for each dataset of varying sizes. For simplicity, I am showing the plot for fitted curve only for the polynomial of degree n=1 and n=9. Plots for other degrees of polynomial can be found in '**Part3**' folder of the Assignment submitted.

[**N.B.: The stopping condition for the gradient descent algorithm is maintained by limiting the number of iterations. I have used 5000 iterations of the algorithm in Part 3 of the question. The weights are initialized randomly from a uniform distribution U(0,1) before the gradient descent algorithm.]**

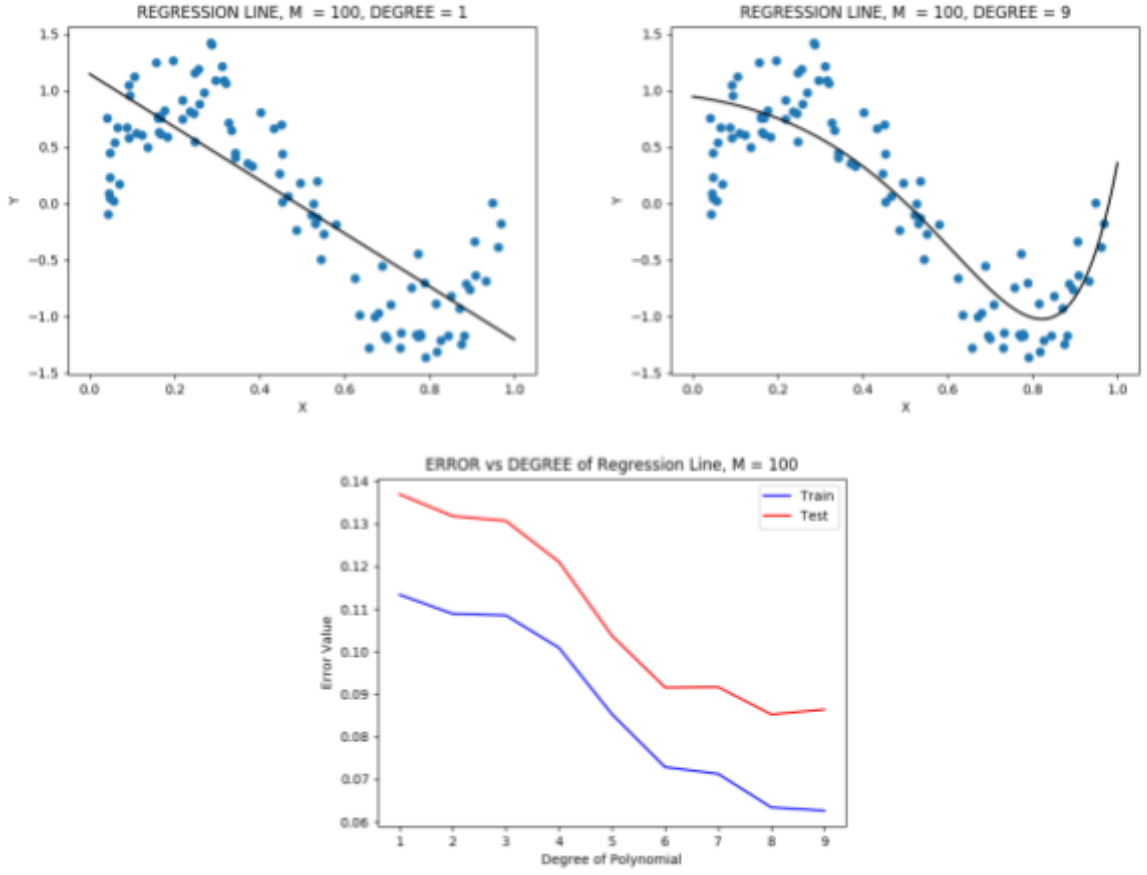


Fig. 5: Polynomial curve and the its error on training set and test set for data of size, M=100

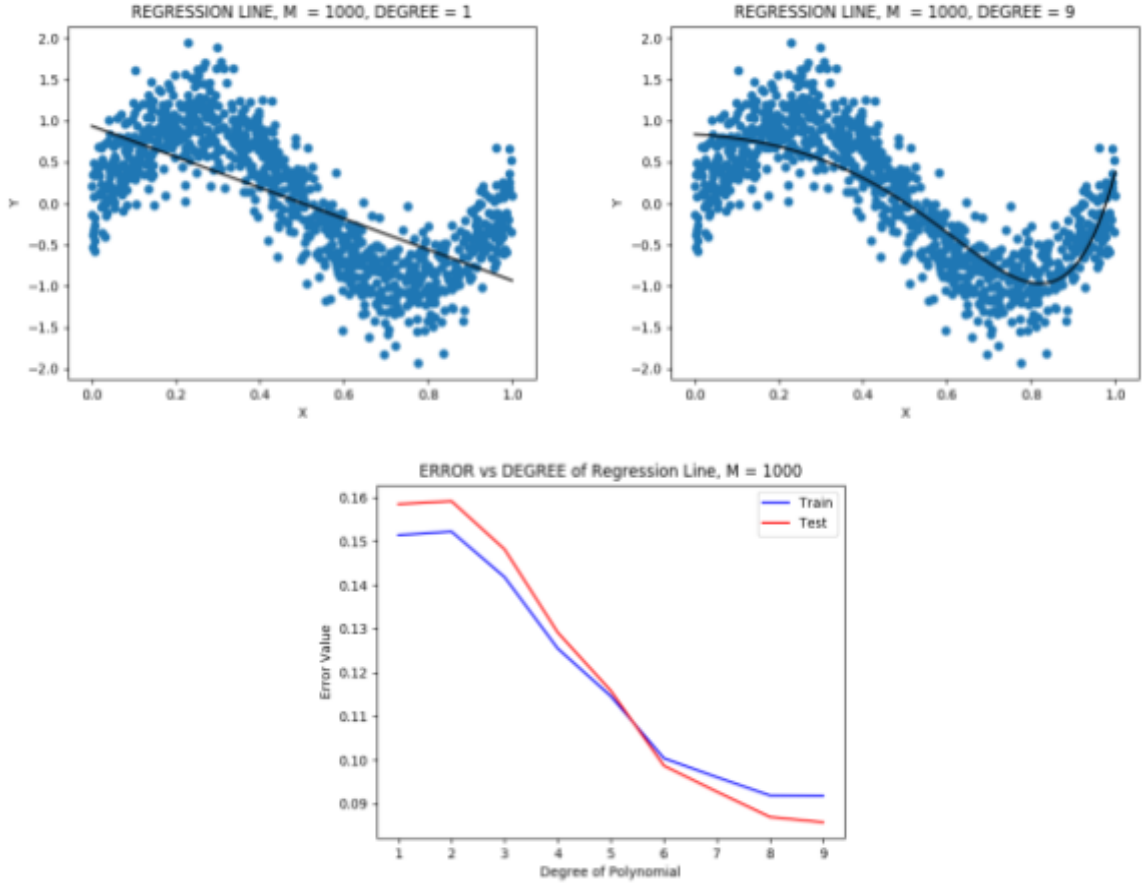


Fig. 6: Plot showing the curve fit and its error vs degree of polynomial when data size, M=1000

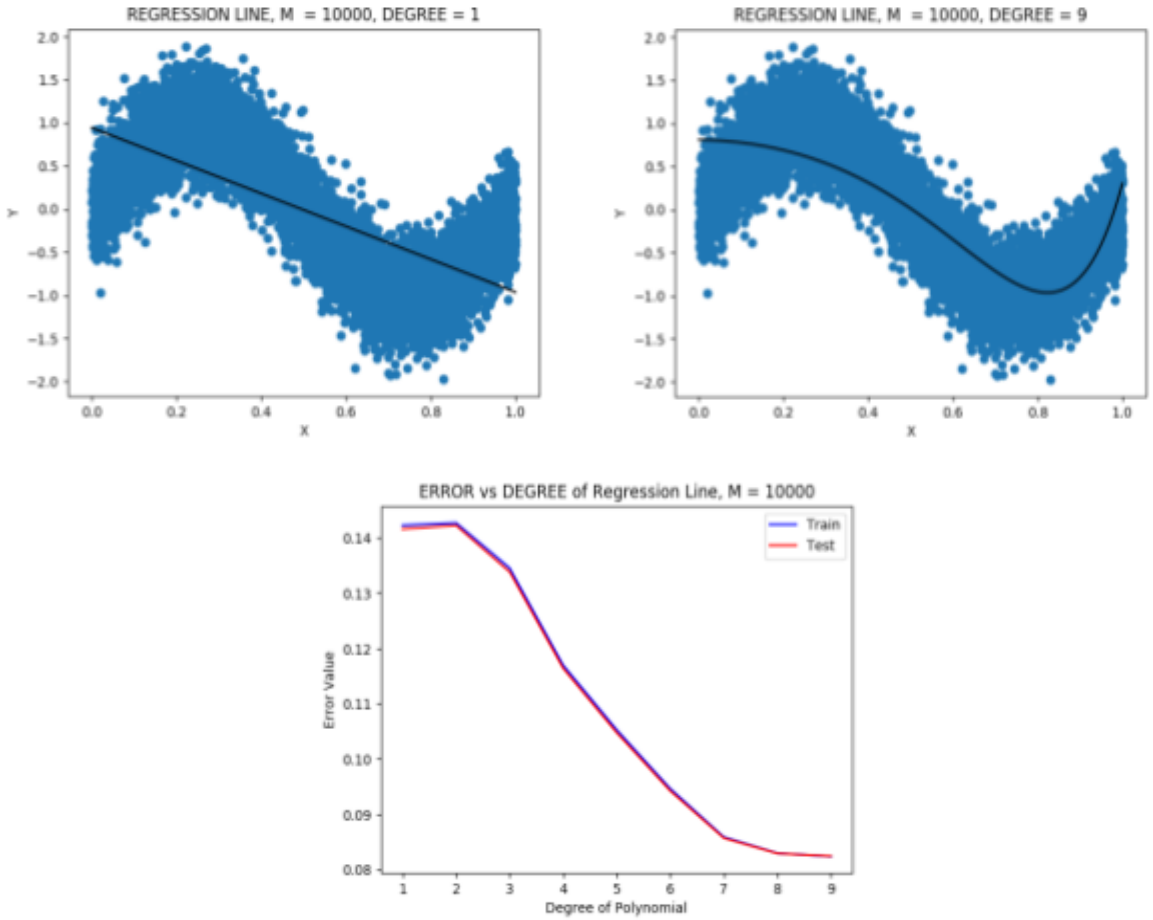


Fig. 7: Plot showing the curve fit and its error vs the degree of polynomial when the data size, M=10000

The value of the weights learned for the varying data sizes are as below:

M=100

Degree of Polynomial n: 1

Weights Learned:

[[1.1469657 -2.35104744]]

Training Error: 0.11334590523292046

Test Error: 0.13692324612705684

Degree of Polynomial n: 2

Weights Learned:

[[0.94209037 -1.07729612 -1.29386077]]

Training Error: 0.10885221126407711

Test Error: 0.13179925786311628

Degree of Polynomial n: 3

Weights Learned:

[[0.98846897 -1.34052752 -1.14940674 0.1471009]]

Training Error: 0.10850075094346788
Test Error: 0.13069014484710909

Degree of Polynomial n: 4
Weights Learned:
[[1.01792074 -1.29649586 -1.58500709 -0.28649624 1.0105697]]
Training Error: 0.1008918527898793
Test Error: 0.12107603471739699

Degree of Polynomial n: 5
Weights Learned:
[[0.99876701 -0.87673031 -2.21551598 -1.04494827 0.50080184 1.88937817]]
Training Error: 0.08526547148764947
Test Error: 0.10369036767293043

Degree of Polynomial n: 6
Weights Learned:
[[0.96169577 -0.5793981 -2.19692693 -1.61812884 -0.2002751   0.91616781
2.35477675]]
Training Error: 0.07283545034563324
Test Error: 0.09154288563344344

Degree of Polynomial n: 7
Weights Learned:
[[0.98874746 -0.71129985 -1.9916133 -1.83939457 0.0693586   0.84977679
0.95900406 1.48811191]]
Training Error: 0.07124745877728732
Test Error: 0.0916373166209235

Degree of Polynomial n: 8
Weights Learned:
[[0.93606591 -0.41691975 -2.07799293 -1.71870665 -0.33873986 -0.02359384
0.70665891 1.4070973   1.70419484]]
Training Error: 0.0633698175999817 3
Test Error: 0.08521942894512258

Degree of Polynomial n: 9
Weights Learned:
[[0.94636855 -0.5270787 -1.83941582 -1.41994666 -1.12828191 0.31354024
0.73156843 0.76163397 1.14114838 1.38064701]]
Training Error: 0.06260067693440273
Test Error: 0.0863697422636331

M=1000

Degree of Polynomial n: 1

Weights Learned:

[[0.93620726 -1.87012158]]

Training Error: 0.1514138265929249

Test Error: 0.158511455156169

Degree of Polynomial n: 2

Weights Learned:

[[0.84644811 -1.36273203 -0.49224191]]

Training Error: 0.1522373594144609

Test Error: 0.1591805931939179

Degree of Polynomial n: 3

Weights Learned:

[[0.93772071 -1.53436431 -1.43790873 1.2386829]]

Training Error: 0.1418138412346168

Test Error: 0.1481778787857076

Degree of Polynomial n: 4

Weights Learned:

[[0.97178971 -1.25160735 -2.28781489 0.18420476 1.85487131]]

Training Error: 0.12548988975798747

Test Error: 0.12916083482812477

Degree of Polynomial n: 5

Weights Learned:

[[0.98015754 -1.15565892 -2.073145   -0.65130774 0.5974172   1.98945991]]

Training Error: 0.11456749009669694

Test Error: 0.11581593782774077

Degree of Polynomial n: 6

Weights Learned:

[[0.90945494 -0.55703748 -2.56682563 -0.96811692 -0.19611437 1.35019883
1.95249256]]

Training Error: 0.1003694563860205

Test Error: 0.09866947283706332

Degree of Polynomial n: 7

Weights Learned:

[[0.89212029 -0.4860971 -2.35534664 -1.03559837 -0.40332346 0.18886562
1.68776009 1.59928349]]

Training Error: 0.09604069689164571
Test Error: 0.0927035775094965

Degree of Polynomial n: 8
Weights Learned:
[[0.84816796 -0.22872545 -2.44381987 -1.07643044 -0.73424724 0.18636306
1.07645143 1.23436211 1.38013267]]
Training Error: 0.09184930274598267
Test Error: 0.08693203928462787

Degree of Polynomial n: 9
Weights Learned:
[[0.83539134 -0.33416669 -1.63517953 -2.00485273 -0.47653464 0.29323414
0.31982405 0.93977777 1.05544125 1.3860587]]
Training Error: 0.09177974306836163
Test Error: 0.08572970394751843

M=10000
Degree of Polynomial n: 1
Weights Learned:
[[0.94178598 -1.90583678]]
Training Error: 0.14231845092540266
Test Error: 0.14164699075783543

Degree of Polynomial n: 2
Weights Learned:
[[0.86634299 -1.47559467 -0.41769823]]
Training Error: 0.142732420055681
Test Error: 0.1422970299960611

Degree of Polynomial n: 3
Weights Learned:
[[0.96178033 -1.73129029 -1.1011926 1.03638838]]
Training Error: 0.1345751413524488
Test Error: 0.13389747868072058

Degree of Polynomial n: 4
Weights Learned:
[[0.96605201 -1.30933026 -1.926764 -0.38506981 2.07730961]]
Training Error: 0.11700739153758026

Test Error: 0.1164363155852868

Degree of Polynomial n: 5
Weights Learned:
[[0.97015126 -1.0998487 -2.11406369 -0.85970847 0.90800775 1.84087057]]
Training Error: 0.10531561778252516
Test Error: 0.10478976339042137

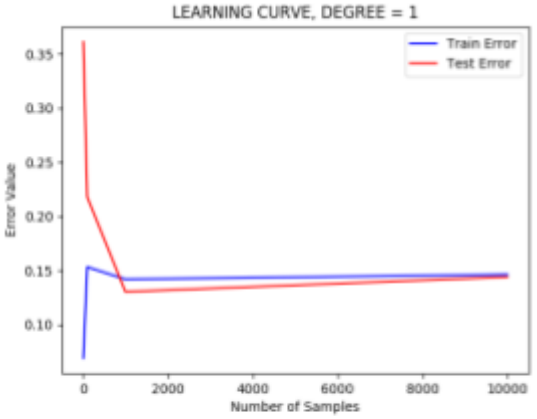
Degree of Polynomial n: 6
Weights Learned:
[[0.9332286 -0.82347163 -2.07311901 -1.00362533 -0.1874127   0.67832821
2.34504956]]
Training Error: 0.0946705711683686
Test Error: 0.09426820610163536

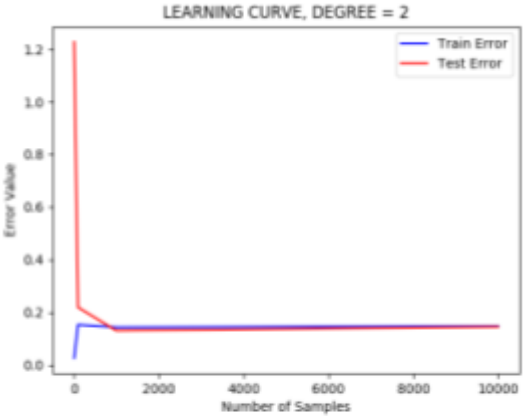
Degree of Polynomial n: 7
Weights Learned:
[[0.86660413 -0.46172704 -1.87041405 -1.67614788 -0.61713524 0.38445422
1.38606823 2.07579573]]
Training Error: 0.08591841817599724
Test Error: 0.08573033417597481

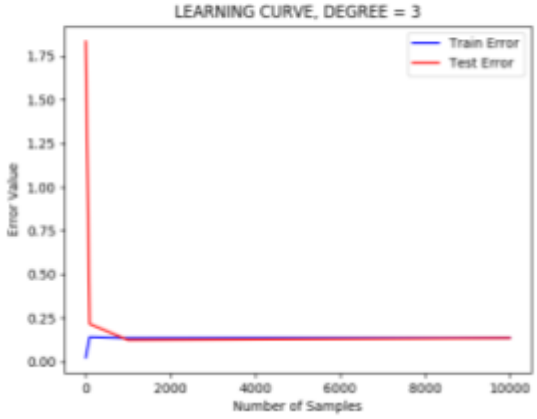
Degree of Polynomial n: 8
Weights Learned:
[[0.82823231 -0.2223425 -2.01643635 -1.78609213 -0.54047932 0.40473019
0.84445897 0.93193355 1.77404923]]
Training Error: 0.0829945802247337
Test Error: 0.08294900977999568

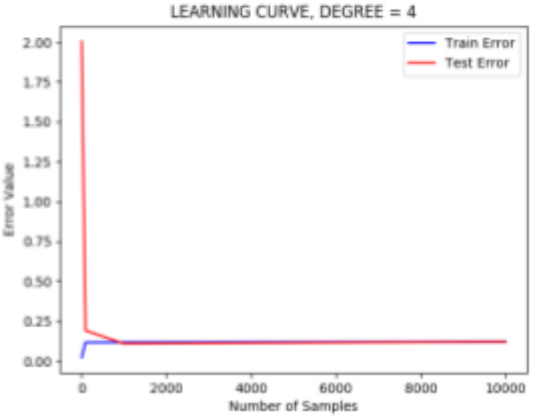
Degree of Polynomial n: 9
Weights Learned:
[[0.80797281 -0.08066807 -2.34178694 -1.17382051 -0.76260833 -0.1235554
0.90421731 0.95487468 1.05378037 1.06218393]]
Training Error: 0.08240545550579866
Test Error: 0.08245511368843397

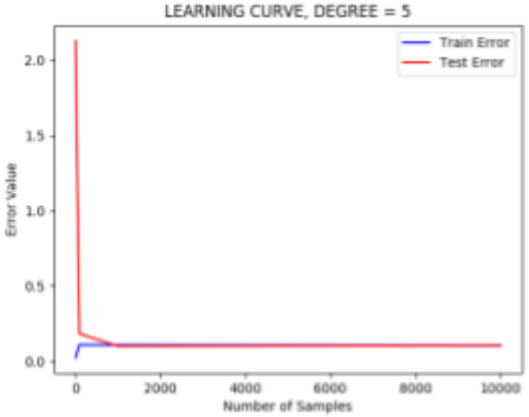
b) The **learning curves** (Train and Test error with respect to the Number of Samples) for different degree of polynomial was generated and is shown as below:

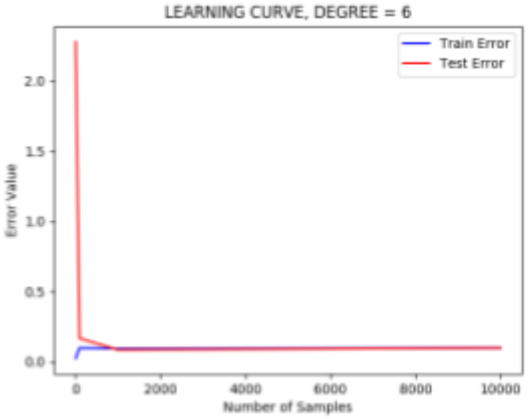

LEARNING CURVE, DEGREE = 1
Train Error
Test Error
Error Value
Number of Samples

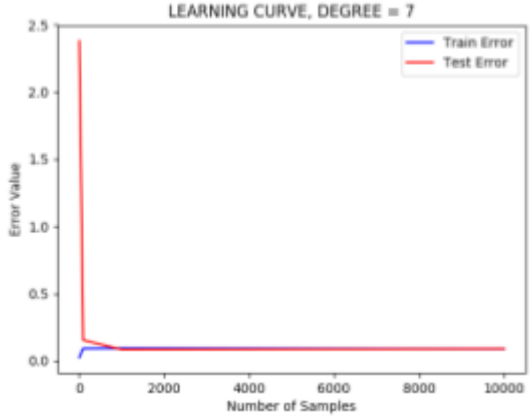

LEARNING CURVE, DEGREE = 2
Train Error
Test Error
Error Value
Number of Samples

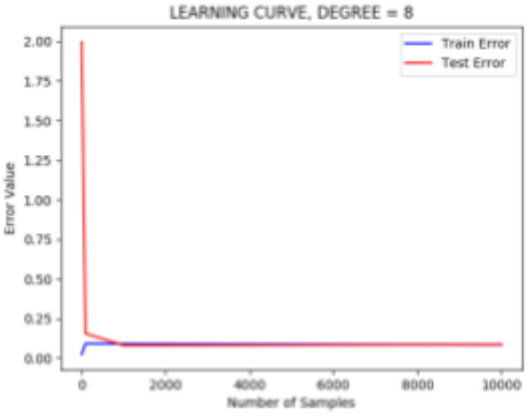

LEARNING CURVE, DEGREE = 3
Train Error
Test Error
Error Value
Number of Samples


LEARNING CURVE, DEGREE = 4
Train Error
Test Error
Error Value
Number of Samples


LEARNING CURVE, DEGREE = 5
Train Error
Test Error
Error Value
Number of Samples


LEARNING CURVE, DEGREE = 6
Train Error
Test Error
Error Value
Number of Samples


LEARNING CURVE, DEGREE = 7
Train Error
Test Error
Error Value
Number of Samples


LEARNING CURVE, DEGREE = 8
Train Error
Test Error
Error Value
Number of Samples

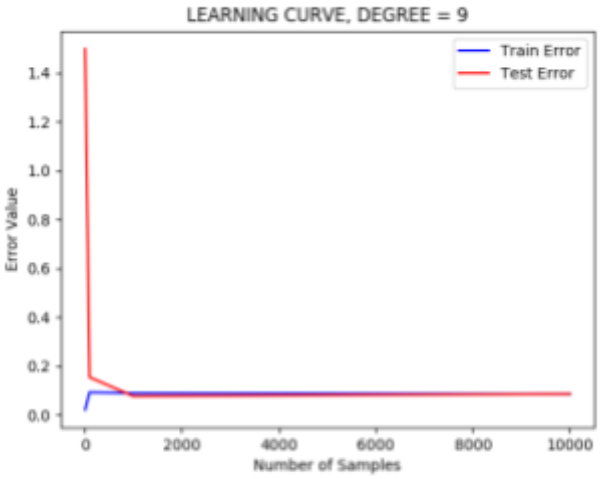


Fig. 8: Learning Curve for different values of degree of polynomial

Thus we have learning curves for different degree of polynomials. We see that initially the test error is very high and train error is low. It is evident, since as we increase number of samples but keep the degree same, the train error increases and test error decreases. But when the number of samples is very high (here, 10000), both the error lines converge.

[**N.B.: 'output.txt' contains the value of the final learned weight parameters that has been reported in this section**]

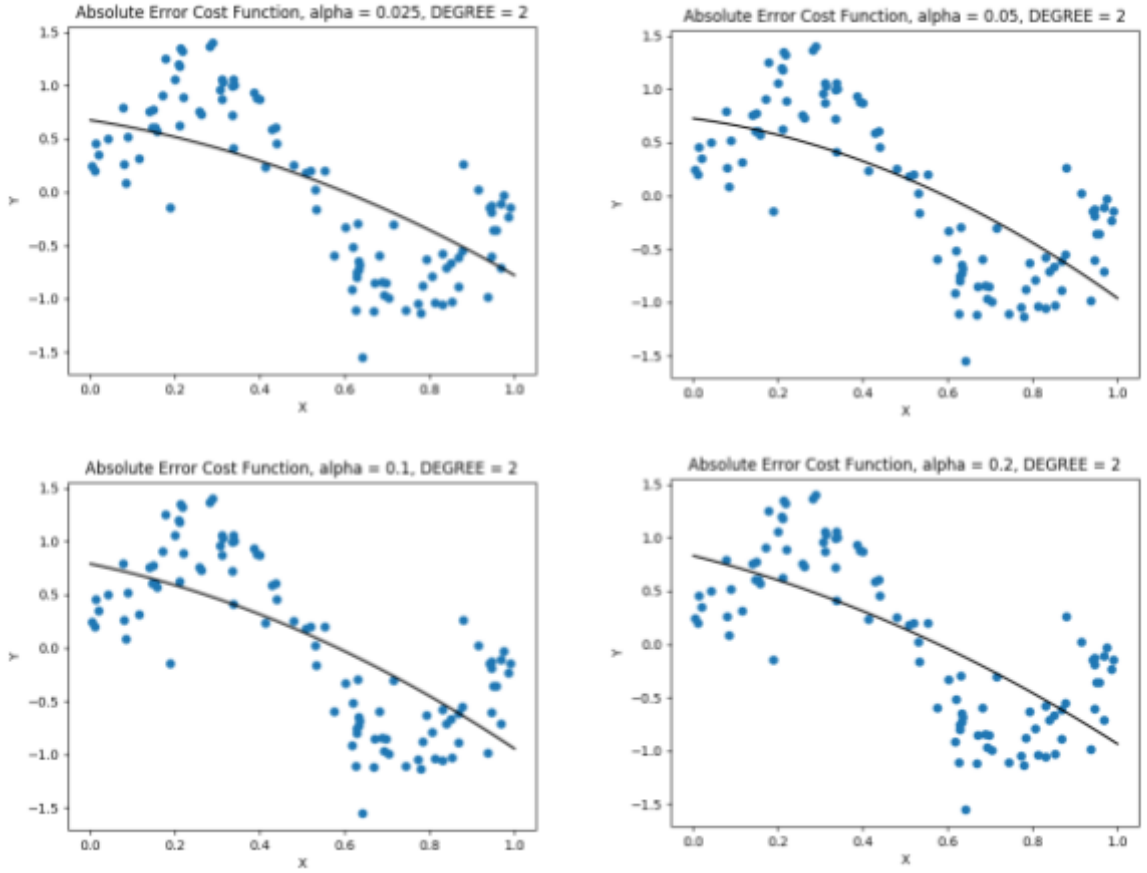
---

4. Experimenting with Cost Functions

a) Two different cost functions i) Mean Absolute Error Function, ii) 4th Power Error Function, were used to minimize the cost in gradient descent. Learning Rate was varied in the range [0.025, 0.05, 0.1, 0.2, 0.5] to show the variance of the curve fitted with the learning rate. The plots are shown below. For simplicity, I am attaching only the curves having the degree of n=2, and n=9. Plots for other degree of polynomial are uploaded along with the assignment.

[**N.B.: Dataset size is taken as 100 since it is not mentioned in this part. The weights are initialized with uniform random number U(0,1) for Mean Absolute Cost Function while 0 for 4th Power Cost Function. Number of iterations run are 5000. Plots that are not provided in this report are uploaded in the zip file.**]

ABSOLUTE MEAN COST FUNCTION (degree = 2)



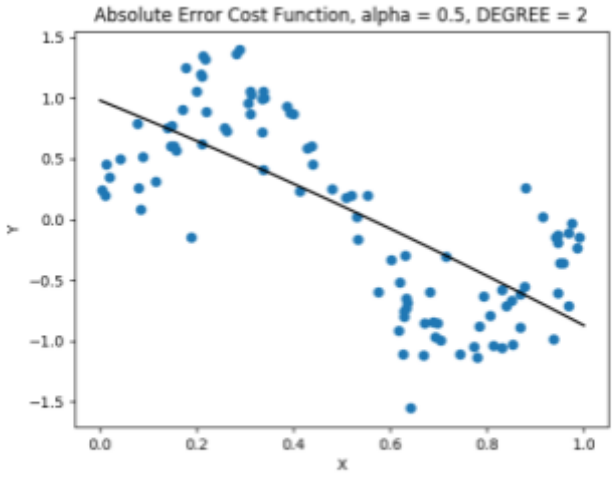
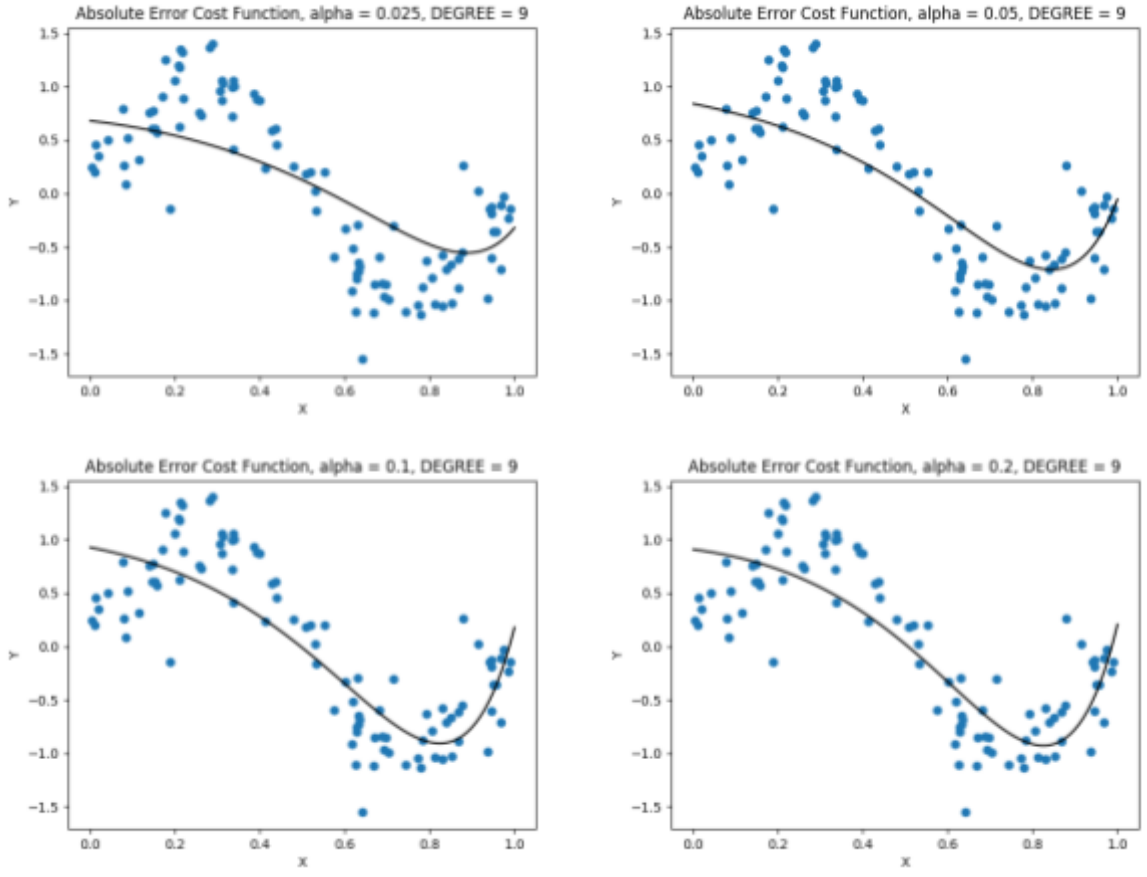


Fig. 9: Plots of Regression curve of degree 2 of various learning rates

ABSOLUTE MEAN COST FUNCTION (degree = 9)



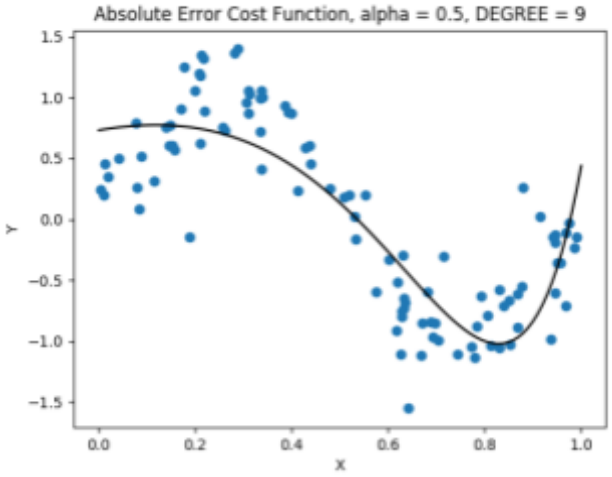
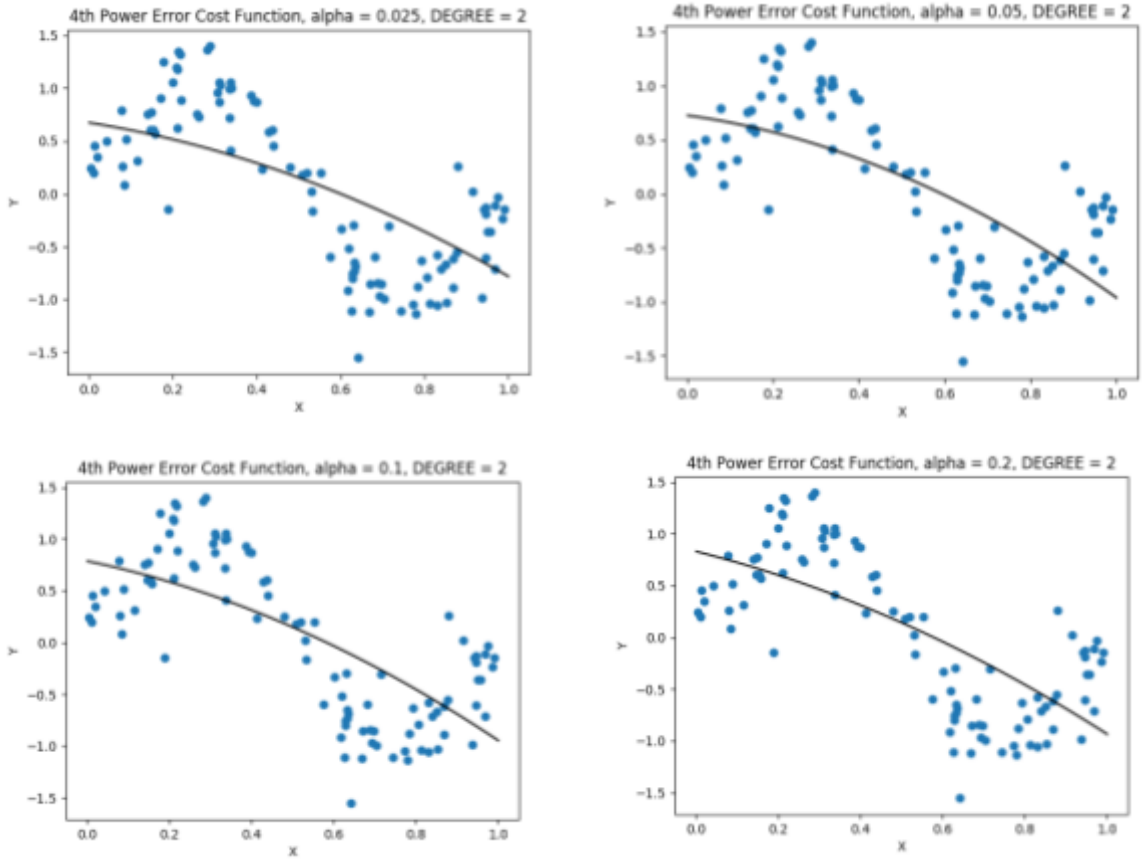


Fig. 10: Plots of Regression Curves of degree 9 for varying learning rates

4th POWER COST FUNCTION (degree = 2)



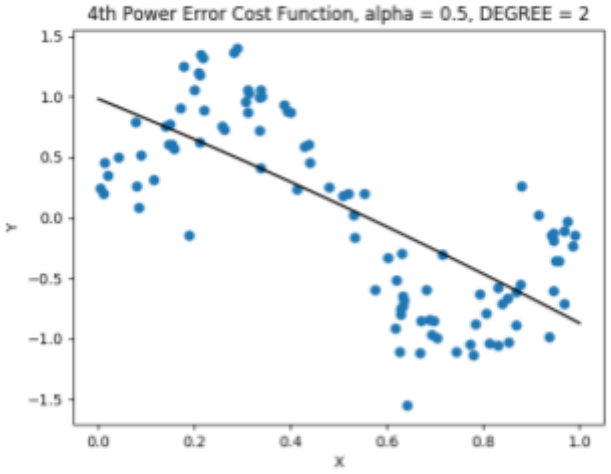
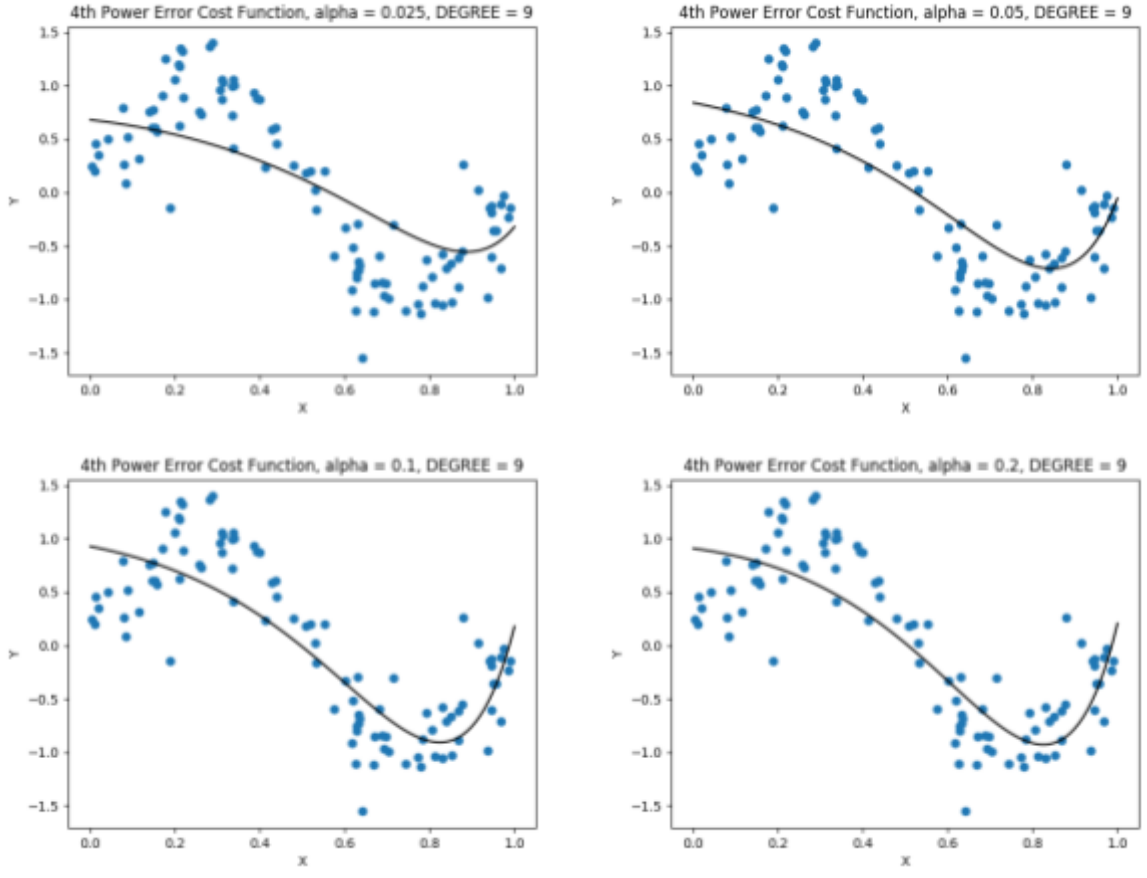


Fig. 11: Plots of regression curves of degree 2 for varying learning rates

4th POWER COST FUNCTION (degree = 9)



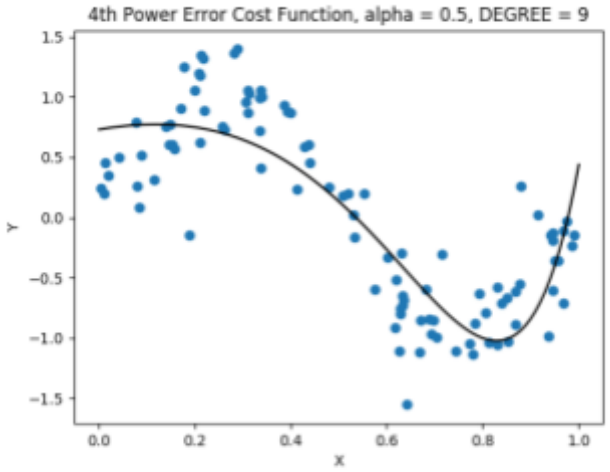


Fig. 12: Plots of regression curves of degree 9 for varying learning rates

We see that with increase in learning rates the fit gets better. This is expected since the convergence is limited with number of iterations and hence a greater learning rate decreases the cost function more.

IMPORTANT: However, we should note that for the **4th Power Error Cost function**, the **initial value of the weight parameters greatly affect the convergence of the gradient descent algorithm at higher learning rates**. This is because the algorithm oscillates and **diverges out**. Hence a slight change of initial weight parameters may be sensitive to the convergence and an error of "**overflow**" may come while running the code.

The weights learned for this part are:

Learning Rate=0.025

Degree of Polynomial n: 1
Absolute Cost Weights Learned:
[[0.56234785 -1.00056632]]
Absolute Cost Test Error: 0.609388744081
4th Power Cost Weights Learned:
[[0.8772942 -1.7073916]]
4th Power Cost Test Error: 0.51768464555

Degree of Polynomial n: 2
Absolute Cost Weights Learned:
[[0.67338258 -0.62488786 -0.83060068]]
Absolute Cost Test Error: 0.549058498884
4th Power Cost Weights Learned:
[[0.75987038 -0.93390504 -0.78927427]]

4th Power Cost Test Error: 0.524314169154

Degree of Polynomial n: 3
Absolute Cost Weights Learned:
[[0.64860032 -0.50919997 -0.27700153 -0.80965482]]
Absolute Cost Test Error: 0.565348925827
4th Power Cost Weights Learned:
[[0.71182851 -0.86201745 -0.6426251 -0.16708308]]
4th Power Cost Test Error: 0.533985396353

Degree of Polynomial n: 4
Absolute Cost Weights Learned:
[[0.68890682 -0.54017261 -0.68031823 -0.58074294 0.2442628]]
Absolute Cost Test Error: 0.545086578352
4th Power Cost Weights Learned:
[[0.71899994 -0.90627299 -0.6900201 -0.21058411 0.19046772]]
4th Power Cost Test Error: 0.528151045847

Degree of Polynomial n: 5
Absolute Cost Weights Learned:
[[0.71048776 -0.61469594 -0.70307183 -0.47106325 -0.32335997 0.63637298]]
Absolute Cost Test Error: 0.530329066741
4th Power Cost Weights Learned:
[[0.74134126 -0.94419304 -0.76147984 -0.30032056 0.09077454 0.36836832]]
4th Power Cost Test Error: 0.514046575368

Degree of Polynomial n: 6
Absolute Cost Weights Learned:
[[0.74480105 -0.92968924 -0.43241805 -0.29436192 -0.42237922 0.21671177
0.40187521]]
Absolute Cost Test Error: 0.52352543846
4th Power Cost Weights Learned:
[[0.76218524 -0.96039285 -0.81629177 -0.38118099 -0.00648297 0.26127546
0.43955906]]
4th Power Cost Test Error: 0.498532349305

Degree of Polynomial n: 7
Absolute Cost Weights Learned:
[[0.65203519 -0.43851002 -0.87115167 -0.52628619 0.03449605 -0.22932267
0.17794232 0.54901329]]
Absolute Cost Test Error: 0.524306030421
4th Power Cost Weights Learned:

[[0.7776491 -0.96143105 -0.85142243 -0.44210098 -0.08471764 0.17200512
0.34359711 0.45331861]]
4th Power Cost Test Error: 0.484762565826

Degree of Polynomial n: 8
Absolute Cost Weights Learned:
[[0.75400404 -0.91913653 -0.62627086 -0.37552054 -0.21538584 0.0303724
0.36473006 0.36259554 0.04578797]]
Absolute Cost Test Error: 0.503701794562
4th Power Cost Weights Learned:
[[0.78811753 -0.95466527 -0.87198051 -0.48508685 -0.14340918 0.10292384
0.26791869 0.37367097 0.43848755]]
4th Power Cost Test Error: 0.47363938867

Degree of Polynomial n: 9
Absolute Cost Weights Learned:
[[0.67869562 -0.45176014 -1.136495   -0.25072304 -0.29232839 0.05037093
0.13661895 -0.19861658 0.41778892 0.72321603]]
Absolute Cost Test Error: 0.498748907519
4th Power Cost Weights Learned:
[[0.79485532 -0.94480311 -0.88311252 -0.51461807 -0.18624192 0.05104984
0.21012007 0.31214459 0.37476587 0.41077463]]
4th Power Cost Test Error: 0.465111015195

Learning Rate=0.05
Degree of Polynomial n: 1
Absolute Cost Weights Learned:
[[0.86609762 -1.62112186]]
Absolute Cost Test Error: 0.523627876083
4th Power Cost Weights Learned:
[[0.93906505 -1.81795347]]
4th Power Cost Test Error: 0.50931399652

Degree of Polynomial n: 2
Absolute Cost Weights Learned:
[[0.72480276 -0.55014708 -1.13790909]]
Absolute Cost Test Error: 0.53070187621
4th Power Cost Weights Learned:
[[0.79918748 -1.11223751 -0.63568792]]

4th Power Cost Test Error: 0.520310915131

Degree of Polynomial n: 3
Absolute Cost Weights Learned:
[[0.74461104 -0.75233609 -1.14479944 0.24836452]]
Absolute Cost Test Error: 0.523032681928
4th Power Cost Weights Learned:
[[0.80957483 -1.19789953 -0.6848987   0.18011267]]
4th Power Cost Test Error: 0.517297751589

Degree of Polynomial n: 4
Absolute Cost Weights Learned:
[[0.76870739 -0.92785765 -0.95076644 0.08827701 0.26144051]]
Absolute Cost Test Error: 0.520528481574
4th Power Cost Weights Learned:
[[0.85469168 -1.28203619 -0.87313402 -0.06800989 0.59651924]]
4th Power Cost Test Error: 0.498599895864

Degree of Polynomial n: 5
Absolute Cost Weights Learned:
[[0.74722494 -0.81678932 -0.69552715 -0.71044911 0.13167941 0.60580323]]
Absolute Cost Test Error: 0.510339602693
4th Power Cost Weights Learned:
[[0.88680233 -1.29071569 -1.00470994 -0.28745096 0.32079307 0.74876374]]
4th Power Cost Test Error: 0.477623627634

Degree of Polynomial n: 6
Absolute Cost Weights Learned:
[[0.74479473 -0.76419488 -0.96858129 -0.35901091 -0.06036209 0.16039161
  0.65080962]]
Absolute Cost Test Error: 0.500710108773
4th Power Cost Weights Learned:
[[0.90284502 -1.26000378 -1.07364644 -0.43659722 0.11694061 0.50980845
  0.76668409]]
4th Power Cost Test Error: 0.460262681214

Degree of Polynomial n: 7
Absolute Cost Weights Learned:
[[0.85688985 -0.81147623 -1.38538117 -0.81484053 0.28663385 -0.1916064
  0.88698327 1.03536236]]
Absolute Cost Test Error: 0.44646589853
4th Power Cost Weights Learned:

[[0.90894708 -1.21863665 -1.10402948 -0.53012593 -0.02152639 0.34148463
0.57938897 0.72632367]]
4th Power Cost Test Error: 0.447673030387

Degree of Polynomial n: 8
Absolute Cost Weights Learned:
[[0.97011499 -1.17316627 -1.44567992 -0.38149024 0.08397973 0.53888697
-0.0087199   0.54797949 0.78815111]]
Absolute Cost Test Error: 0.436434676292
4th Power Cost Weights Learned:
[[0.91006263 -1.17945104 -1.11445827 -0.58759655 -0.11358592 0.22585907
0.44838139 0.58571765 0.66467232]]
4th Power Cost Test Error: 0.43946398918

Degree of Polynomial n: 9
Absolute Cost Weights Learned:
[[0.83860151 -0.81062871 -0.87389647 -1.3770423 -0.07224858 0.16117852
0.55555847 0.05630798 0.31919036 1.14561906]]
Absolute Cost Test Error: 0.431903417722
4th Power Cost Weights Learned:
[[0.90912884 -1.14637134 -1.11559618 -0.62334444 -0.17548435 0.14577141
0.35616537 0.48572693 0.55997714 0.59747961]]
4th Power Cost Test Error: 0.434688082037

<u>Learning Rate=0.1</u>
Degree of Polynomial n: 1
Absolute Cost Weights Learned:
[[1.0089238 -1.84445963]]
Absolute Cost Test Error: 0.506072516207
4th Power Cost Weights Learned:
[[0.94233797 -1.82377137]]
4th Power Cost Test Error: 0.508926830892

Degree of Polynomial n: 2
Absolute Cost Weights Learned:
[[0.78787589 -0.81730599 -0.91688856]]
Absolute Cost Test Error: 0.523558701744
4th Power Cost Weights Learned:
[[0.85670666 -1.4086526 -0.36425128]]
4th Power Cost Test Error: 0.515380489776

Degree of Polynomial n: 3
Absolute Cost Weights Learned:
[[0.82314476 -1.03001767 -1.08180937 0.43157606]]
Absolute Cost Test Error: 0.512010058681
4th Power Cost Weights Learned:
[[0.92654843 -1.62079207 -0.77590837 0.67645961]]
4th Power Cost Test Error: 0.500492966311

Degree of Polynomial n: 4
Absolute Cost Weights Learned:
[[0.94167246 -1.2353526 -1.31632386 0.12553701 0.76130524]]
Absolute Cost Test Error: 0.481860050578
4th Power Cost Weights Learned:
[[0.97151816 -1.60986939 -1.13278117 0.06649209 1.08689217]]
4th Power Cost Test Error: 0.476905931098

Degree of Polynomial n: 5
Absolute Cost Weights Learned:
[[0.93855274 -1.13704621 -1.38851014 -0.45759581 0.51668091 1.00451786]]
Absolute Cost Test Error: 0.459720449285
4th Power Cost Weights Learned:
[[0.98100382 -1.49372325 -1.2986452 -0.32496726 0.54364177 1.16064149]]
4th Power Cost Test Error: 0.456507202691

Degree of Polynomial n: 6
Absolute Cost Weights Learned:
[[0.97888397 -1.12924997 -1.63571701 -0.92235448 0.54674486 0.7467389
  1.28212199]]
Absolute Cost Test Error: 0.432558784453
4th Power Cost Weights Learned:
[[0.97573518 -1.37356823 -1.35685602 -0.544819   0.20865062 0.74974536
  1.10015099]]
4th Power Cost Test Error: 0.44209844606

Degree of Polynomial n: 7
Absolute Cost Weights Learned:
[[1.0262945 -1.31166582 -1.15592771 -1.25815862 -0.04332817 0.71477639
  0.76376876 1.2012243]]
Absolute Cost Test Error: 0.414994713164
4th Power Cost Weights Learned:

[[0.96691391 -1.27690048 -1.36846184 -0.66624169 0.00539379 0.49083891
0.80531212 0.99337013]]
4th Power Cost Test Error: 0.433501598473

Degree of Polynomial n: 8
Absolute Cost Weights Learned:
[[0.93134573 -0.91352258 -1.3892305 -1.05738148 -0.47318889 0.24953627
0.23514625 1.46185361 1.01632756]]
Absolute Cost Test Error: 0.406417363426
4th Power Cost Weights Learned:
[[0.95816657 -1.20189016 -1.36061093 -0.73448978 -0.12206776 0.32179906
0.60852343 0.77887526 0.86963192]]
4th Power Cost Test Error: 0.428601385574

Degree of Polynomial n: 9
Absolute Cost Weights Learned:
[[0.92709128 -0.77266899 -1.58298433 -1.27295598 -0.25908005 -0.20542928
0.71254825 0.80488848 0.94377431 0.88248159]]
Absolute Cost Test Error: 0.396639409735
4th Power Cost Weights Learned:
[[0.94953514 -1.14189957 -1.34251784 -0.77083673 -0.20308719 0.20801802
0.47181535 0.62662392 0.70721388 0.73933004]]
4th Power Cost Test Error: 0.425294037245

Learning Rate=0.2
Degree of Polynomial n: 1
Absolute Cost Weights Learned:
[[1.00972873 -1.84619391]]
Absolute Cost Test Error: 0.505963625027
4th Power Cost Weights Learned:
[[0.9423455 -1.82378475]]
4th Power Cost Test Error: 0.508925946907

Degree of Polynomial n: 2
Absolute Cost Weights Learned:
[[0.82979484 -0.98844821 -0.77762679]]
Absolute Cost Test Error: 0.519219006637
4th Power Cost Weights Learned:
[[0.93474601 -1.82950458 0.02818464]]
4th Power Cost Test Error: 0.510230429448

Degree of Polynomial n: 3
Absolute Cost Weights Learned:
[[0.999628 -1.54324252 -1.09744424 0.90362937]]
Absolute Cost Test Error: 0.492888366744
4th Power Cost Weights Learned:
[[1.01450865 -1.9258607 -1.00301003 1.22377885]]
4th Power Cost Test Error: 0.490068187234

Degree of Polynomial n: 4
Absolute Cost Weights Learned:
[[1.02003313 -1.43183881 -1.61283638 -0.06873618 1.55248033]]
Absolute Cost Test Error: 0.461070868907
4th Power Cost Weights Learned:
[[1.01736886 -1.64755109 -1.5311257 0.0853064 1.58066385]]
4th Power Cost Test Error: 0.46533978347

Degree of Polynomial n: 5
Absolute Cost Weights Learned:
[[1.0182003 -1.11266044 -2.02259975 -0.94052999 0.47220901 2.43811905]]
Absolute Cost Test Error: 0.426615263162
4th Power Cost Weights Learned:
[[0.99522711 -1.372456 -1.70098165 -0.49714988 0.70446632 1.581997]]
4th Power Cost Test Error: 0.444845472239

Degree of Polynomial n: 6
Absolute Cost Weights Learned:
[[1.00310741 -0.97149391 -1.72243714 -1.24396444 -0.25377255 0.79687649
2.33338451]]
Absolute Cost Test Error: 0.405209140849
4th Power Cost Weights Learned:
[[0.97213695 -1.17429211 -1.72885505 -0.7814437 0.22405483 0.96752875
1.44813235]]
4th Power Cost Test Error: 0.431653059751

Degree of Polynomial n: 7
Absolute Cost Weights Learned:
[[0.93196881 -0.66159033 -1.93564964 -1.3156284 -0.37349278 0.34392589
1.26810207 1.80052892]]
Absolute Cost Test Error: 0.394647151329
4th Power Cost Weights Learned:

[[0.95210737 -1.03474592 -1.70391829 -0.91979534 -0.04823453 0.59931386
1.01539531 1.25540827]]
4th Power Cost Test Error: 0.423990081353

Degree of Polynomial n: 8
Absolute Cost Weights Learned:
[[1.0171995 -1.00955201 -1.52653475 -1.35822022 -0.60629119 0.67833038
0.76071889 0.96493843 1.20469907]]
Absolute Cost Test Error: 0.393095598454
4th Power Cost Weights Learned:
[[0.93466937 -0.93652181 -1.65669236 -0.97798572 -0.20135091 0.37350163
0.73720457 0.94076221 1.03506802]]
4th Power Cost Test Error: 0.419107680362

Degree of Polynomial n: 9
Absolute Cost Weights Learned:
[[0.90841962 -0.52437071 -1.71404256 -1.50407496 -0.74908647 0.57949523
0.27354268 1.06999374 0.97973626 0.88569459]]
Absolute Cost Test Error: 0.386435184979
4th Power Cost Weights Learned:
[[0.92089566 -0.87449945 -1.60545359 -0.99012991 -0.27834721 0.24264076
0.56450454 0.73647551 0.80733008 0.81393632]]
4th Power Cost Test Error: 0.416392207437

Learning Rate=0.5

Degree of Polynomial n: 1
Absolute Cost Weights Learned:
[[1.01154101 -1.84772133]]
Absolute Cost Test Error: 0.505898110657
4th Power Cost Weights Learned:
[[0.9423455 -1.82378475]]
4th Power Cost Test Error: 0.508925946903

Degree of Polynomial n: 2
Absolute Cost Weights Learned:
[[0.9812825 -1.62635572 -0.22830737]]
Absolute Cost Test Error: 0.50739033622
4th Power Cost Weights Learned:

[[1.02432167 -2.34127345 0.51676884]]
4th Power Cost Test Error: 0.506993190436

Degree of Polynomial n: 3
Absolute Cost Weights Learned:
[[1.00637808 -1.29938133 -2.46615177 2.09168817]]
Absolute Cost Test Error: 0.472166126895
4th Power Cost Weights Learned:
[[1.03017426 -1.78770028 -1.80296699 1.94367125]]
4th Power Cost Test Error: 0.480871475

Degree of Polynomial n: 4
Absolute Cost Weights Learned:
[[1.07415548 -1.00526795 -3.1573427 -0.40181553 3.29804423]]
Absolute Cost Test Error: 0.433627283867
4th Power Cost Weights Learned:
[[0.97737218 -1.12727458 -2.53209375 -0.14028964 2.43232811]]
4th Power Cost Test Error: 0.448021606834

Degree of Polynomial n: 5
Absolute Cost Weights Learned:
[[0.98629193 -0.54016687 -2.82761583 -1.66490918 0.65117653 3.27828988]]
Absolute Cost Test Error: 0.396160704703
4th Power Cost Weights Learned:
[[0.92726272 -0.6980225 -2.63723958 -1.0253982   0.91902539 2.39261616]]
4th Power Cost Test Error: 0.423943828833

Degree of Polynomial n: 6
Absolute Cost Weights Learned:
[[0.81153455 0.27506423 -3.05891833 -1.90337189 -0.58687776 1.64139264
2.86273908]]
Absolute Cost Test Error: 0.375315081825
4th Power Cost Weights Learned:
[[0.88831797 -0.44004064 -2.5515399 -1.38347328 0.16477592 1.34647597
2.09820496]]
4th Power Cost Test Error: 0.410688976728

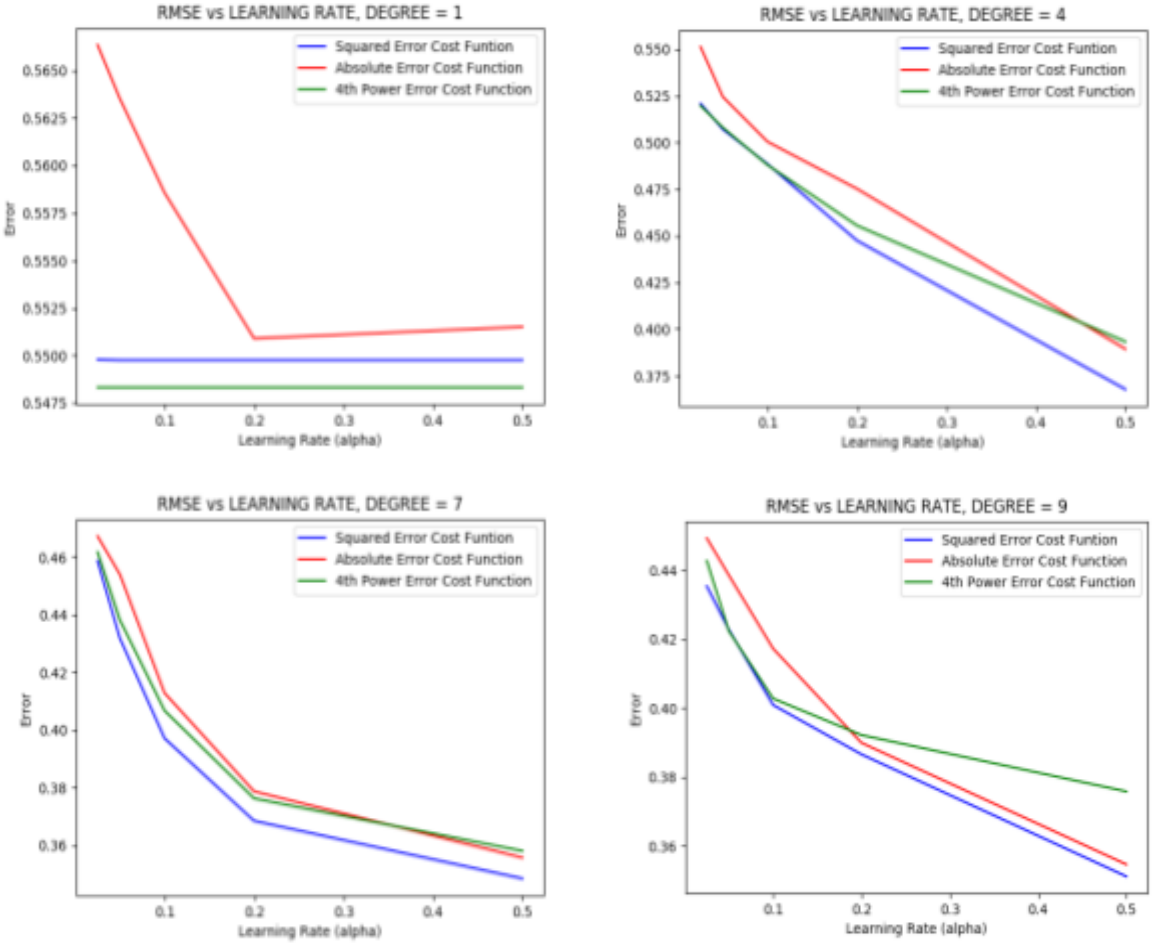
Degree of Polynomial n: 7
Absolute Cost Weights Learned:
[[0.76560019 0.4757578 -2.49769559 -2.78164876 -0.6296921   0.88610068
1.43194482 2.54414179]]
Absolute Cost Test Error: 0.364318129197

4th Power Cost Weights Learned:
[[0.85982786 -0.29880178 -2.40852089 -1.48932232 -0.19929323 0.77360604
1.36994233 1.67421929]]
4th Power Cost Test Error: 0.403435586753

Degree of Polynomial n: 8
Absolute Cost Weights Learned:
[[0.72530407 0.77702473 -2.82568529 -2.06036844 -0.94631113 0.00490567
1.16333809 1.48237549 2.04336863]]
Absolute Cost Test Error: 0.364014428553
4th Power Cost Weights Learned:
[[0.8435997 -0.24789751 -2.27539594 -1.4749574 -0.34014099 0.49184015
0.97251139 1.18549495 1.22060909]]
4th Power Cost Test Error: 0.400744507687

Degree of Polynomial n: 9
Absolute Cost Weights Learned:
[[0.73164653 0.67077867 -2.31578205 -2.75533248 -0.82303872 0.33884897
1.0213062   0.92678916 0.92255178 1.72144445]]
Absolute Cost Test Error: 0.364819735848
4th Power Cost Weights Learned:
[[0.83780075 -0.25300754 -2.18110282 -1.42022263 -0.36602081 0.38143919
0.78557244 0.93292321 0.91316088 0.79414564]]
4th Power Cost Test Error: 0.400866933585

b) An instance of the plot of RMSE vs Learning Rate is shown below. The plot shows only 4 degrees of the polynomial curve for simplicity. Plots for other degrees of polynomial are uploaded in zip file.



Here, we see that the curve fitted with squared error cost function works better on test set than the other two on an average. However this varies with dataset and there is no conclusive support to use one over other. But, we see that the error decreases with increase in learning rate for the data generated. Hence, **we should use a learning rate of 0.5 in this dataset**. Infact, in average, almost in all the cases, **a higher learning rate (0.5) gives a lower RMSE.** It is to be noted that the plot doesn't vary as much for n=1 degree of the fit, as expected.

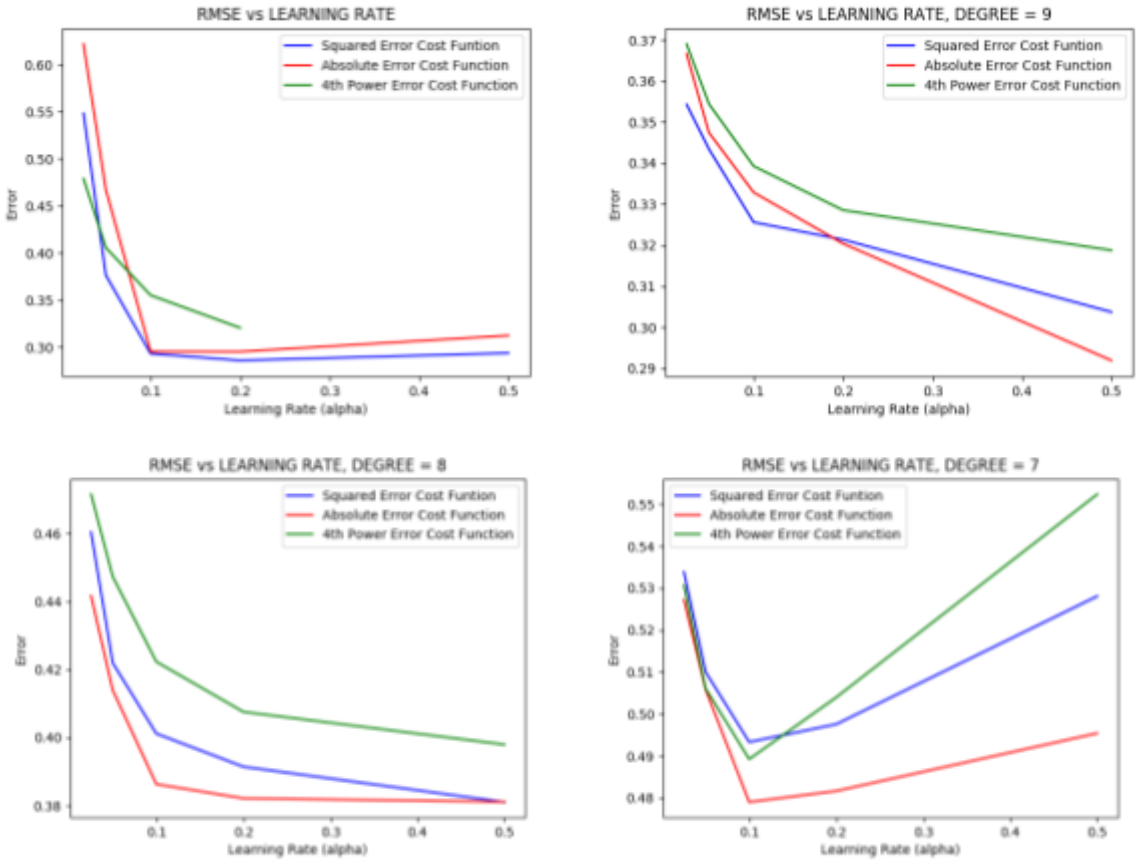


Fig. 14: RMSE vs Learning Rate for various dataset

Lets see some plots that give a general idea regarding the use of learning rates. We see in Fig. 15 that in the first plot, there is no RMSE plotted against learning rate of 0.5 for 4th power Cost function. This is because there is an overflow in the value caused due to divergence in gradient descent. We also see the general trend that a **higher learning rate tends to give a better performance.** Also in the last plot, there is an overshoot.

General Note: Since the problem statement does not mention anything about reproducing the same data again, no random seed is maintained in my code. Hence, the data and plot shown in this report are not reproducible. However, by running the code for a few times, a general trend can be seen which is similar to the theories and plots mentioned in this report. This proves that the work is genuine. The plots and the data mentioned in this report are original and has not been tampered with. The method of running the code is written in the README file attached in the zip file submitted.
