ELL409 Assignment 1

Shravan Nawandar 2019MT10764
Shaurya Mohan 2019MT10658

April 2021

Contents

Chapter 1

Question 1

1.1 Introduction

This is a report on Question 1 of assignment 1 for the course ELL409. It explores 4 different error functions that are tested on different degrees of polynomials and different iterations to observe how $E_{in}(w)$ and $E_{out}(w)$ changes with number of iterations and degree. The estimate of $E_{out}(w)$ is computed using testing error while that of $E_{in}(w)$ is computed using the training error. The first section discusses the types of error functions used in this assignment, it also gives out the mathematical formula used for them. Then we study the behaviour of in-sample error and out-sample error for various degrees and iterations for different types of error function. This helps us to determine good-fit, under-fit and over-fit over various degrees and iterations helping us determine the underlying polynomial. Once the underlying polynomial is determined we are going to use mean square error to estimate the variance of noise. This process is repeated again for regularized error, after finding the optimal regularization parameter. Again we determine our best guess of underlying polynomial corresponding to our optimal regularization parameter. Again the whole process is repeated over a larger part of the complete data set.

1.2 Loss functions implemented

1.2.1 Mean absolute value error function

Mean absolute value error function is calculated based on the formula:

$$\frac{1}{N}\sum_{i=1}^{N}|w^T x_i - y_i|$$

where $x_i = [1, x, x^2, x^3, ..., x^d]$ is the i^{th} training example for a degree d polynomial. Gradient of this error function

$$\nabla_w E_{in}(w^T x) = \frac{1}{N}\sum_{i=1}^{N}(sign(w^T x_i - y_i)x_i)$$

where $sign(x) = 1$ if $x \geq 0$, $sign(x) = -1$ if $x < 0$

1.2.2 Log-cosh error function

Log-cosh error function is calculated based on the formula:

$$\frac{1}{N}\sum_{i=1}^{N} log(cosh(w^T x_i - y_i))$$

Gradient of this error function

$$\nabla_w E_{in}(w^T x) = \frac{1}{N}\sum_{i=1}^{N} tanh(w^T x_i - y_i)x_i$$

where $cosh(x) = \frac{e^x + e^{-x}}{2}$ the hyperbolic cosine function

1.2.3 Log sigmoid error function

Log sigmoid error function is calculated based on the formula:

$$\frac{1}{N}\sum_{i=1}^{N} ln(1 + e^{(w^T x_i - y_i)^2})$$

Gradient of this error function

$$\nabla_w E_{in}(w^T x) = \frac{1}{N}\sum_{i=1}^{N} \frac{2(w^T x_i - y_i)x_i}{1 + e^{-(w^T x_i - y_i)^2}}$$

where ln(x) is natural log of x

1.2.4 Mean Square root error function

Mean square root error function is calculated based on the formula:

$$\frac{1}{N}\sum_{i=1}^{N} |w^T x_i - y_i|^{1/2}$$

Gradient of this error function

$$\nabla_w E_{in}(w^T x) = \frac{1}{N}\sum_{i=1}^{N} \frac{sign(w^T x_i - y_i)x_i}{2|w^T x_i - y_i|^{1/2}}$$

1.3 No regularization - trained on 20 points

1.3.1 Mean absolute Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{-2}	$9.27\ 10^4$	$3.81\ 10^4$
2	10^{-3}	$7.81\ 10^4$	$6.58\ 10^4$
3	10^{-3}	$5.21\ 10^4$	$3.97\ 10^4$
4	10^{-5}	$3.98\ 10^4$	$7.14\ 10^4$
5	10^{-5}	$1.20\ 10^4$	$3.82\ 10^4$
6	10^{-6}	$1.68\ 10^3$	$9.83\ 10^4$
7	10^{-8}	$2.90\ 10^1$	$6.23\ 10^2$
8	10^{-10}	$5.95\ 10^1$	$1.08\ 10^4$
9	10^{-10}	$1.33\ 10^3$	$2.86\ 10^4$

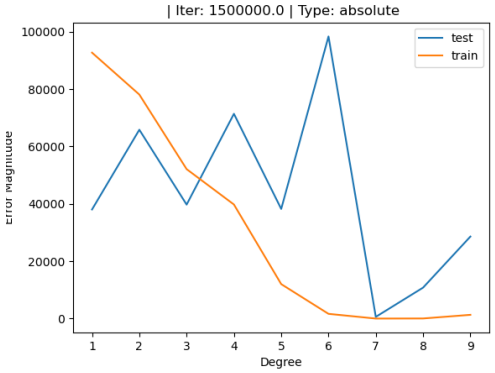


Figure 1.1: 1500000 iterations

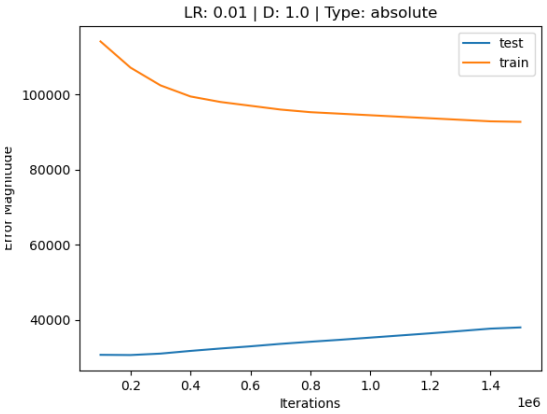


Figure 1.2: Degree 1

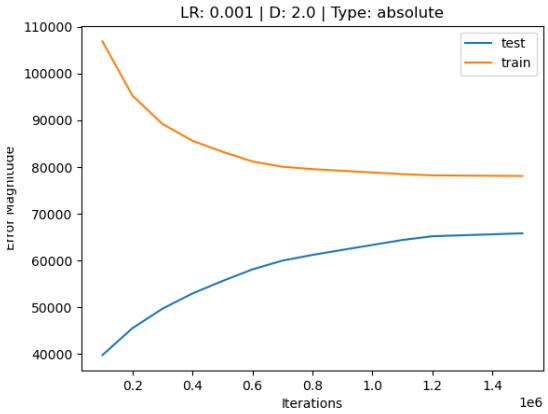


Figure 1.3: Degree 2

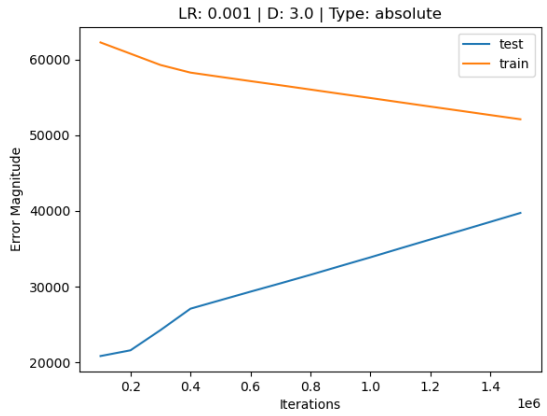


Figure 1.4: Degree 3

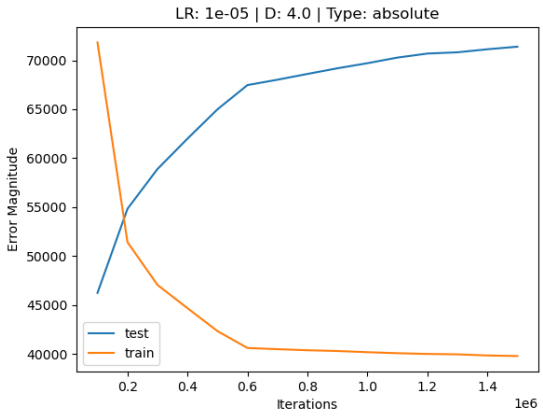


Figure 1.5: Degree 4

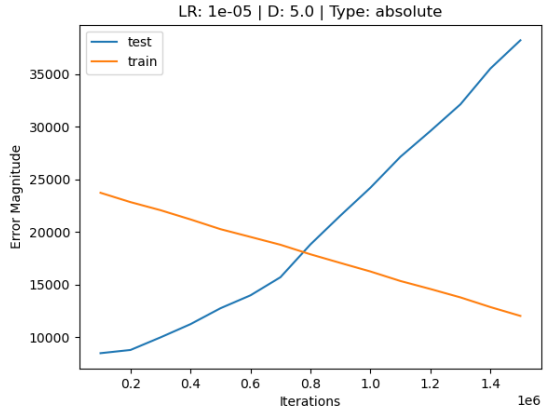


Figure 1.6: Degree 5

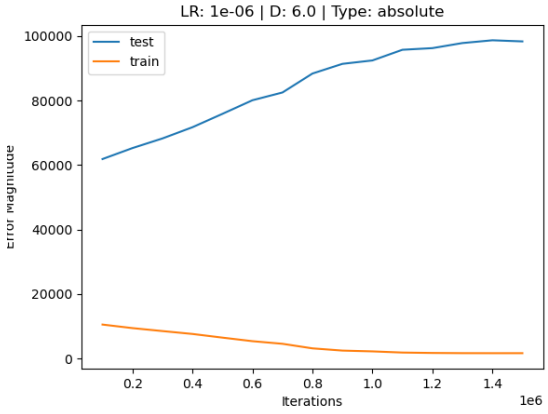


Figure 1.7: Degree 6

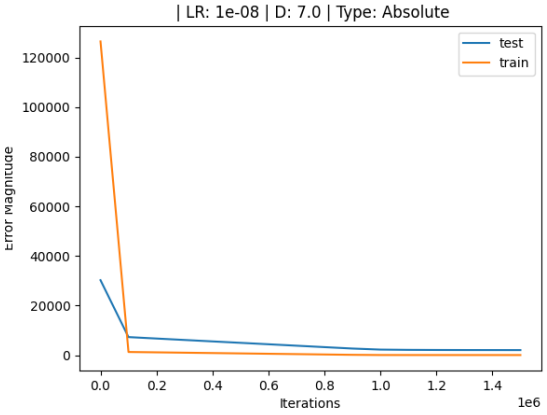


Figure 1.8: Degree 7

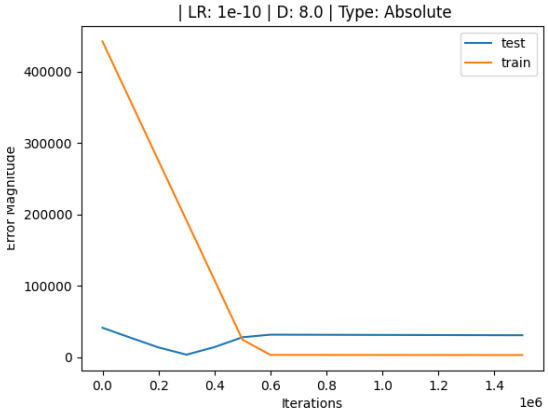


Figure 1.9: Degree 8

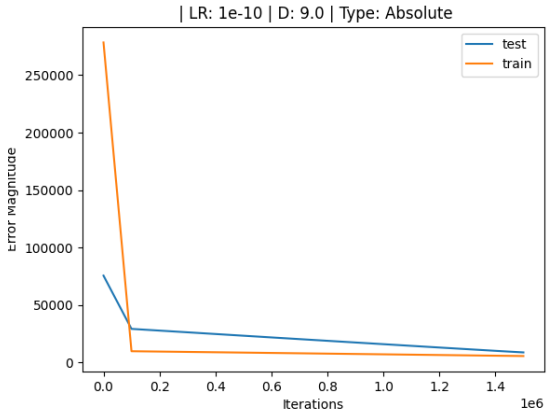


Figure 1.10: Degree 9

Interpretation: Graph 1

- The training error decreases as we increase the degree, hence as we increase the complexity of the model the training error decreases
- High test error for degree 6 signifies that the model is an over-fit.
- The training error flattens at degree 6,7,8 and 9 showing that these complex models have achieved a very good fit on the training data.
- Among degrees 7,8 and 9, for degree 7 we achieve the lowest testing error after which it starts increasing again which implies that degree 8 and 9 are over-fit, hence we can safely consider 7 to be the degree of the polynomial.

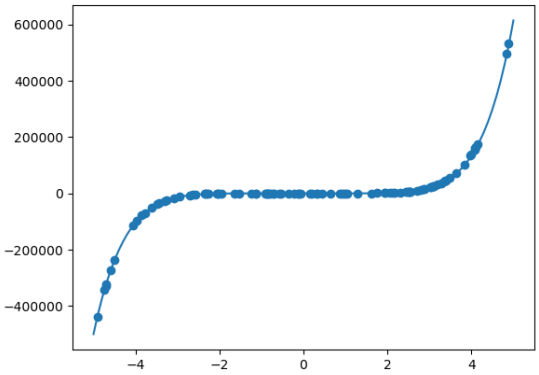
Interpretation: Graphs 2-10

- As we can see for degree 1,2 and 3 the testing error always remains less than training error. Also the magnitude of training error is very high signifying under-fit among these degrees. These aren't very useful for us.
- Then as iterations increase for degree 4,5 and 6 the difference between training and testing error keeps on increasing, as testing error becomes much larger and training error is significantly smaller which implies over-fit on the training data.
- For degree 7,8 and 9 we can observe that the model learns the optimal parameters really quickly. Also these have relatively lower value for training error.

Conclusion:
From the above interpretations we can safely conclude that degree 7 is the best fit for this error function. So now we will use degree 7 for estimating the noise and for guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1400000
Degree: 7
Learning Rate: 10^{-7}
Training Error: 19.448532260627598
Testing Error: 2526.9021938444153
Estimated Noise variance: 472.524
Mean of error: -1.9442

Figure 1.11: The estimated polynomial

Estimated Polynomial $-4.38 * 10^{-2} + 2.11x + 0.71x^2 + 4.68x^3 + 0.61x^4 + 2.18x^5 + 2.88x^6 + 6.88x^7$

1.3.2 Log Sigmoid Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{-6}	$1.11\ 10^{10}$	$1.36\ 10^{10}$
2	10^{-7}	$6.15\ 10^{9}$	$4.51\ 10^{10}$
3	10^{-8}	$3.09\ 10^{9}$	$3.94\ 10^{9}$
4	10^{-7}	$1.52\ 10^{8}$	$1.99\ 10^{11}$
5	10^{-8}	$3.29\ 10^{7}$	$3.53\ 10^{10}$
6	10^{-10}	$4.08\ 10^{6}$	$7.10\ 10^{10}$
7	10^{-15}	$1.45\ 10^{7}$	$6.82\ 10^{7}$
8	10^{-16}	$5.56\ 10^{7}$	$1.26\ 10^{10}$
9	10^{-18}	$2.83\ 10^{10}$	$1.25\ 10^{11}$

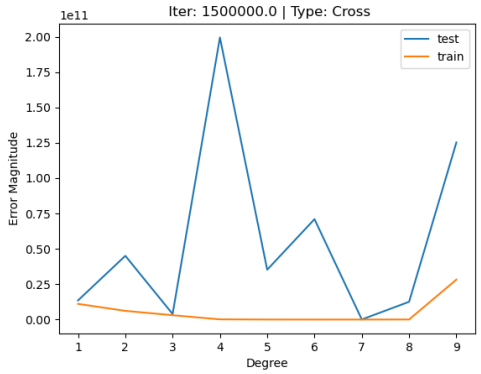


Figure 1.12: 1500000 iterations

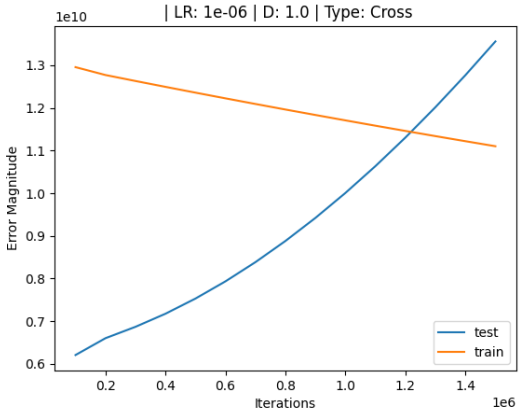


Figure 1.13: Degree 1

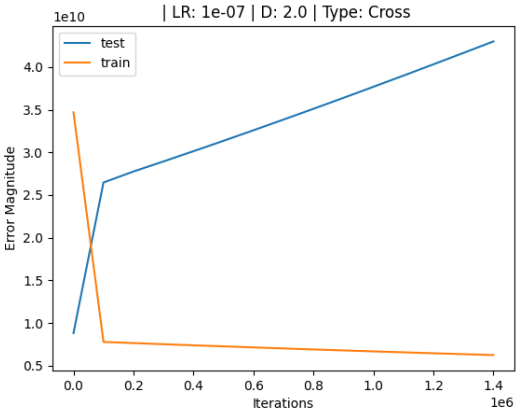


Figure 1.14: Degree 2

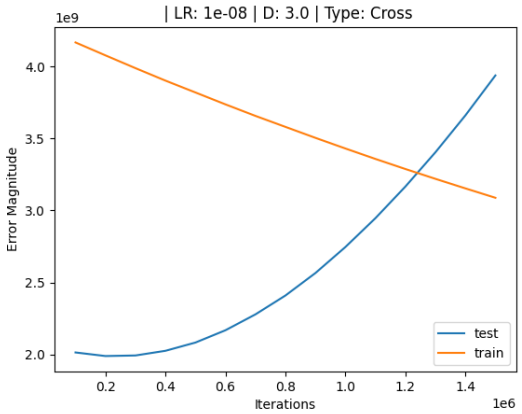


Figure 1.15: Degree 3

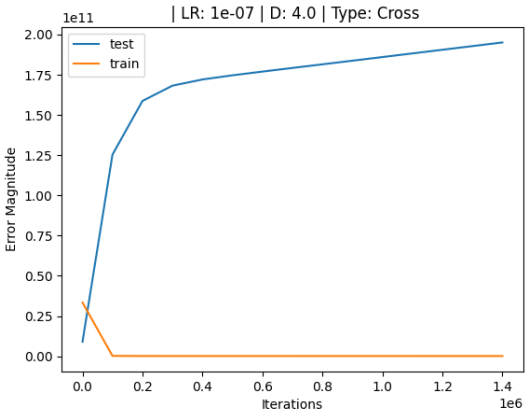


Figure 1.16: Degree 4

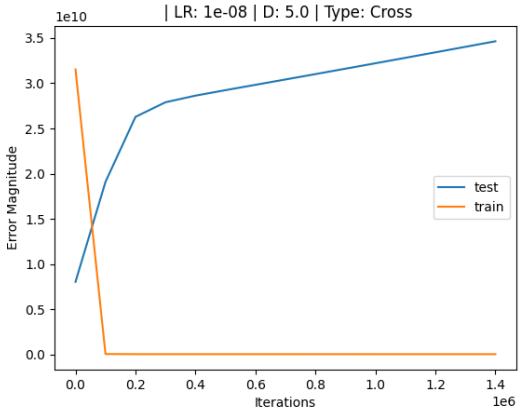


Figure 1.17: Degree 5

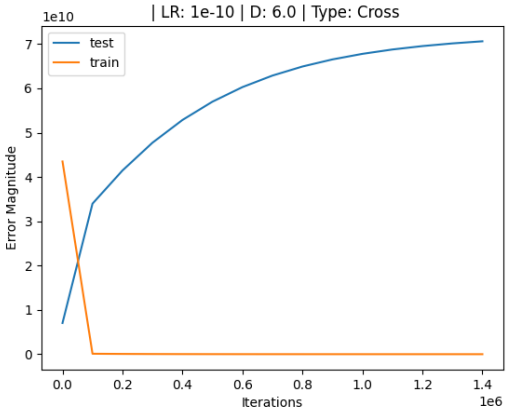


Figure 1.18: Degree 6

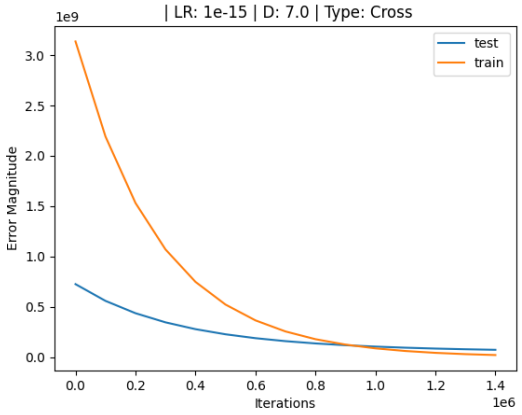


Figure 1.19: Degree 7

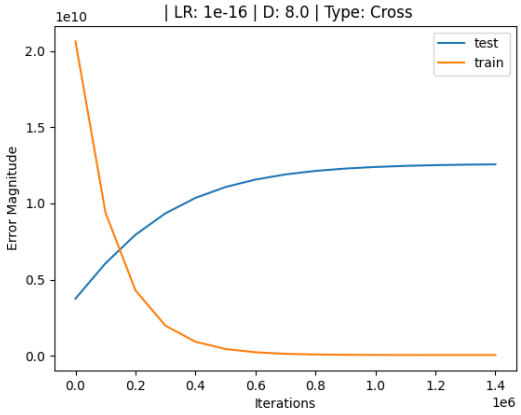


Figure 1.20: Degree 8

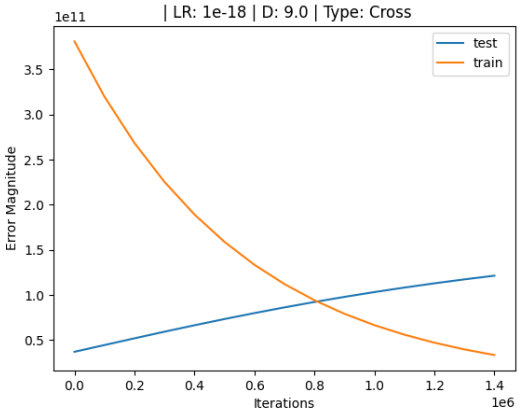


Figure 1.21: Degree 9

Interpretation: Graph 1

- The training error decreases as we increase the degree till degree 7, hence as we increase the complexity of the model the training error decreases.
- The training error starts increasing after degree 7 which means that for the given number of iterations degree 8 and 9 are worse fit than degree 7.
- For degree 1,2 and 3 both the testing and training error are significantly high, signifying a poor fit on entire data set.
- For degree 4,5 and 6 the testing error is really high while training error is low. This signifies a clear over-fit on the training data
- Degree 7 has the lowest testing error (by quite a margin). Hence from this graph degree 7 looks like the underlying polynomial.

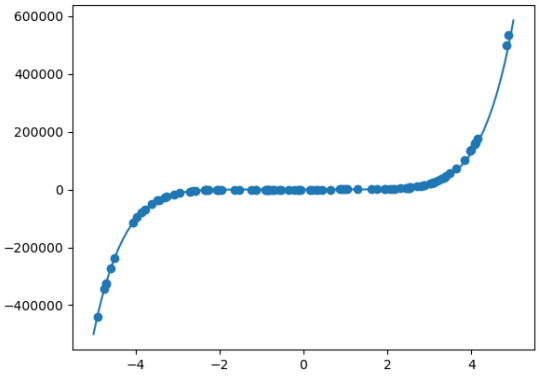
Interpretation: Graphs 2-10

- As we can see for degree 1,2 and 3 the testing error is increasing and seems like will increase further on increasing the number of iterations.
- The training and testing error graph for degree 4,5,6 starts flattening as we reach 1500000 iterations. This means that the model has been trained almost completely. The high testing error signifies that these are an over fit on the training data.
- Only for degree 7 the testing error decreases. While for others they increase.
- For degree 9 the shape of the graph tells us that the training error would have decreased even further, signifying that the model is still not completely trained.

Conclusion:
Again we are getting degree 7 to be the best guess for the polynomial as can be seen from interpretations of graph 1 and graphs 2 - 10.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1000000
Training Error: 1060.2778021
Testing Error: 53729436.08059
Estimated Noise variance: 1060.276
mean difference: 11.428
Degree: 7
Learning Rate: 10^{-9}

Figure 1.22: The estimated polynomial

Polynomial Guessed: $4.96 + 4.09x - 0.23x^2 + 4.56x^3 + 1.80x^4 + 2.04x^5 + 2.63x^6 + 6.84x^7$

1.3.3 Mean Root Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{3}	$1.99\ 10^{2}$	$4.38\ 10^{2}$
2	10^{2}	$1.81\ 10^{2}$	$3.41\ 10^{2}$
3	10^{1}	$1.27\ 10^{2}$	$3.60\ 10^{2}$
4	10^{-1}	$1.17\ 10^{2}$	$2.57\ 10^{2}$
5	10^{-2}	$6.30\ 10^{1}$	$1.90\ 10^{2}$
6	10^{-3}	$3.41\ 10^{1}$	$1.90\ 10^{2}$
7	10^{-6}	$5.84\ 10^{0}$	$3.82\ 10^{1}$
8	10^{-7}	$7.72\ 10^{0}$	$4.51\ 10^{1}$
9	10^{-7}	$1.76\ 10^{1}$	$8.29\ 10^{1}$

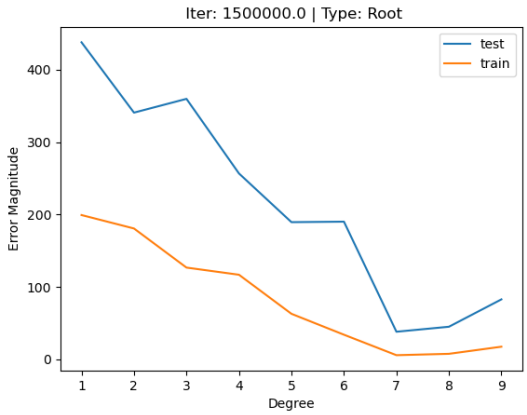


Figure 1.23: 1500000 iterations

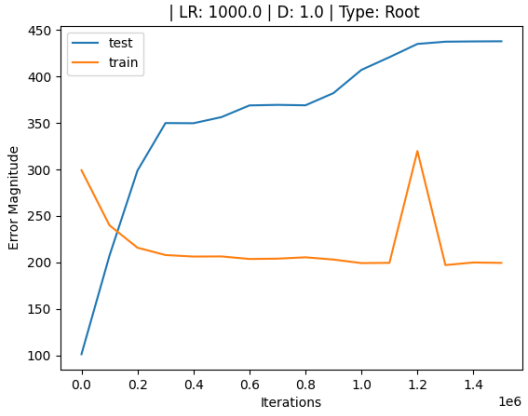


Figure 1.24: Degree 1

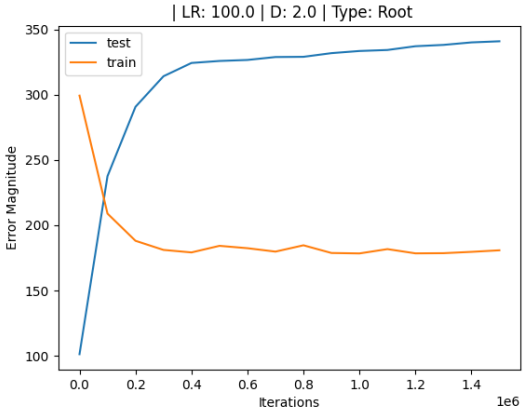


Figure 1.25: Degree 2

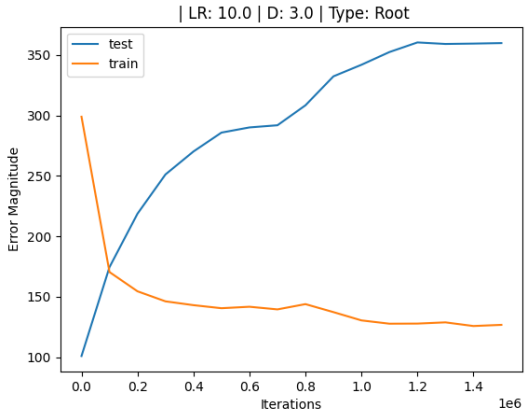


Figure 1.26: Degree 3

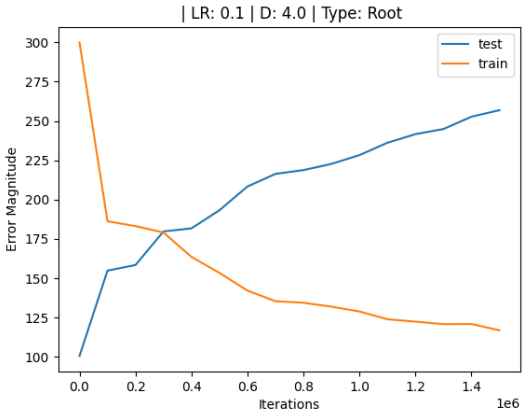


Figure 1.27: Degree 4

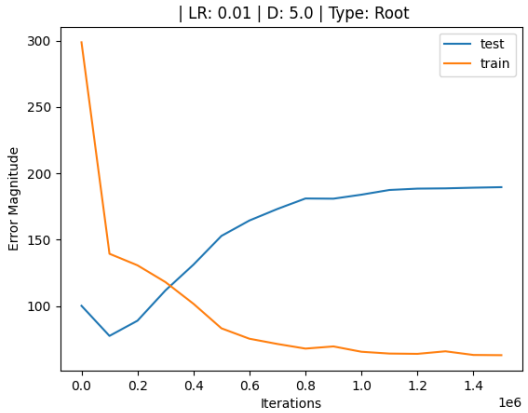


Figure 1.28: Degree 5

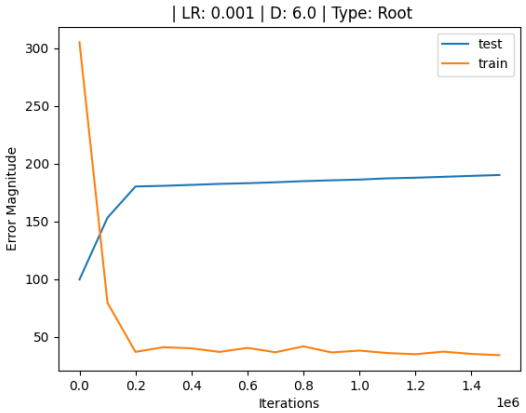


Figure 1.29: Degree 6

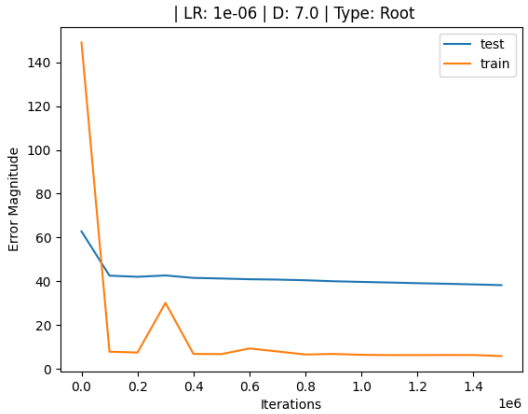


Figure 1.30: Degree 7

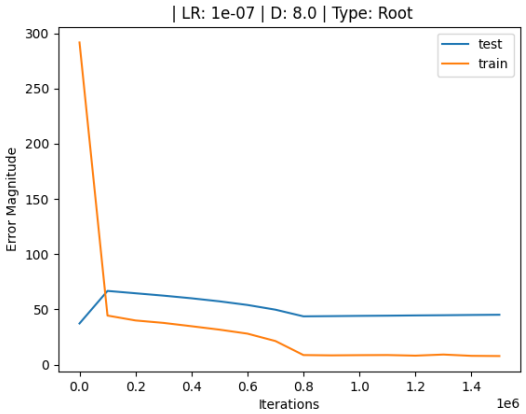


Figure 1.31: Degree 8

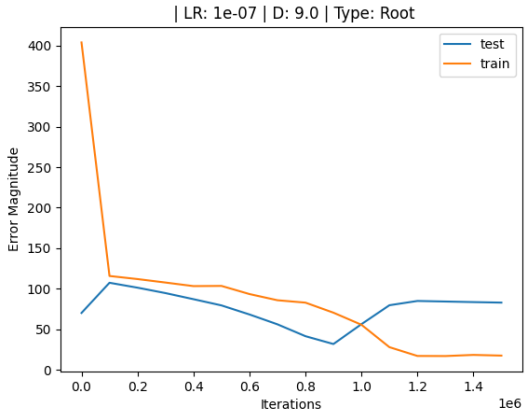


Figure 1.32: Degree 9

Interpretation: Graph 1

- The training error decreases as we increase the degree till degree 7, hence as we increase the complexity of the model the training error decreases
- The Test error also decrease till 7 where it achieves a minima and then increases again.
- For the initial degrees the training error is very high signifying a bad fit on the data.
- After degree 7 the testing error increases hence degree 7 seems to be the best fit.

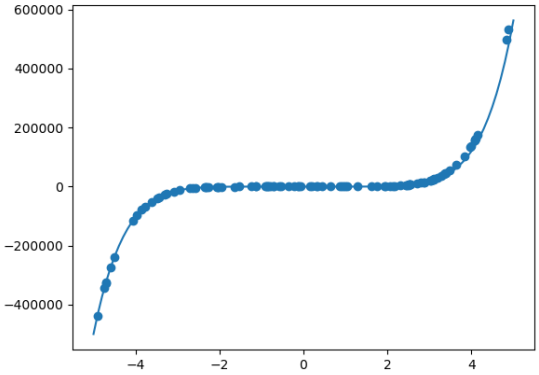
Interpretation: Graphs 2-10

- For all degrees except 4 we can see the both the error curves flatten out signifying that they have been almost completely trained.
- We see a similar shape for degrees 1 to 6 where the testing error starts from a lower value and increases while training error decreases
- For degree 7,8 and 9 the training error starts from a really high value but falls down pretty fast after which they are more or less constant.
- Here we observe the learning rates are larger as compared to the other error functions, this can be attributed to the low value of mean root error function which implies lower gradient hence larger learning rate.

Conclusion:
For this error function too seven comes out to be the best guess. So now we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1500000
Training Error: 5.83593
Testing Error: 38.22695
Estimated Noise variance: 2546.56184
mean difference: 10.1185473
Degree: 7
Learning Rate: 10^{-6}

Figure 1.33: The estimated polynomial

Estimated Polynomial $2.97 + 1.07x + 0.84x^2 + 4.33x^3 + 0.43x^4 + 0.42x^5 + 2.01x^6 + 6.77x^7$

1.3.4 Hyperbolic Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{0}	$6.04\ 10^{4}$	$2.65\ 10^{5}$
2	10^{-1}	$3.75\ 10^{4}$	$2.71\ 10^{5}$
3	10^{-2}	$1.97\ 10^{4}$	$1.82\ 10^{5}$
4	10^{-5}	$3.98\ 10^{4}$	$7.14\ 10^{4}$
5	10^{-5}	$1.21\ 10^{4}$	$3.80\ 10^{4}$
6	10^{-8}	$1.13\ 10^{4}$	$5.82\ 10^{4}$
7	10^{-8}	$1.06\ 10^{2}$	$5.94\ 10^{3}$
8	10^{-10}	$8.73\ 10^{2}$	$2.13\ 10^{4}$
9	10^{-11}	$2.85\ 10^{4}$	$9.04\ 10^{4}$

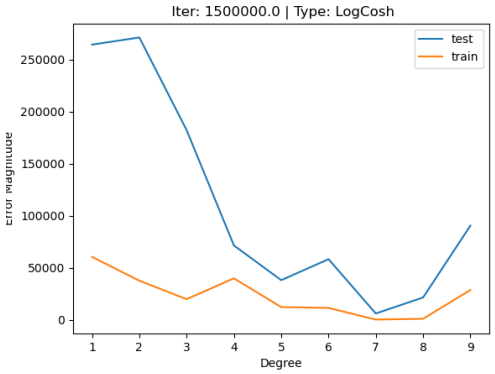


Figure 1.34: 1500000 iterations

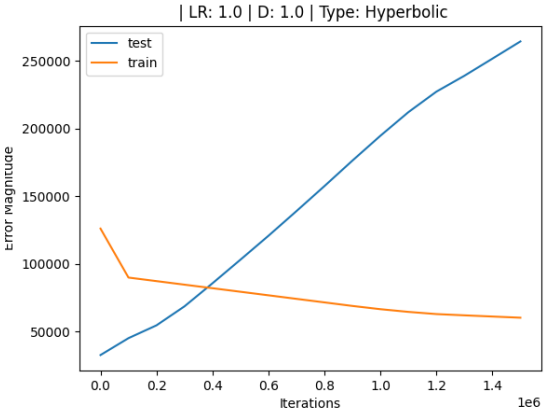


Figure 1.35: Degree 1

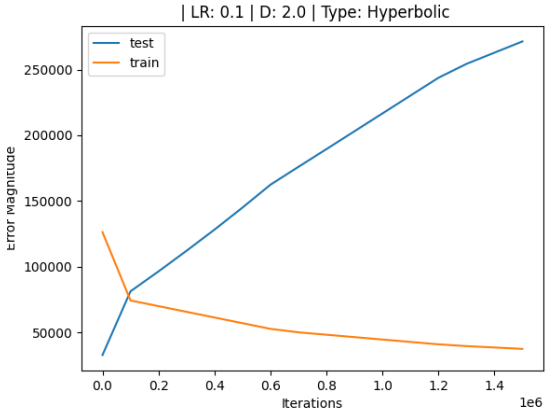


Figure 1.36: Degree 2

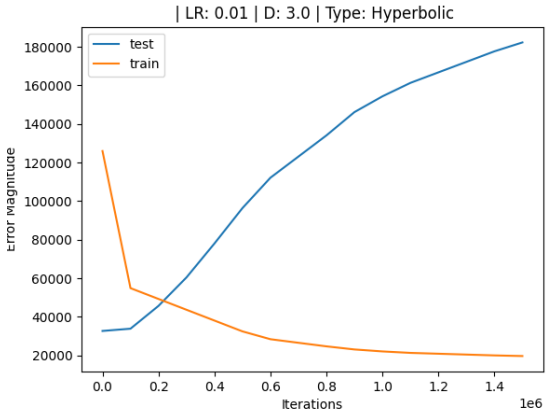


Figure 1.37: Degree 3

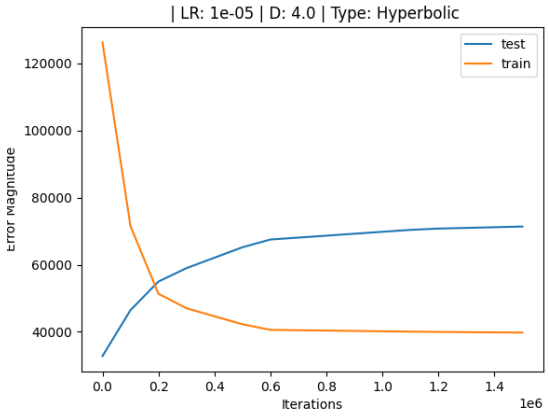


Figure 1.38: Degree 4

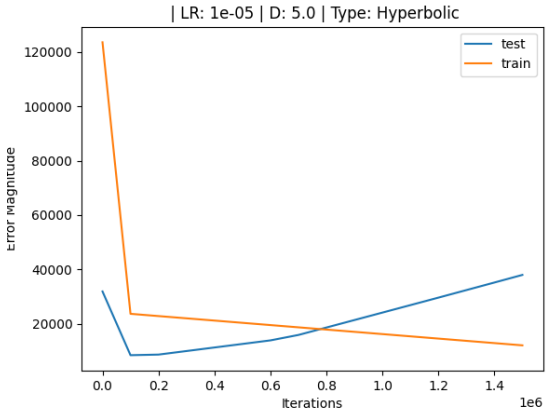


Figure 1.39: Degree 5

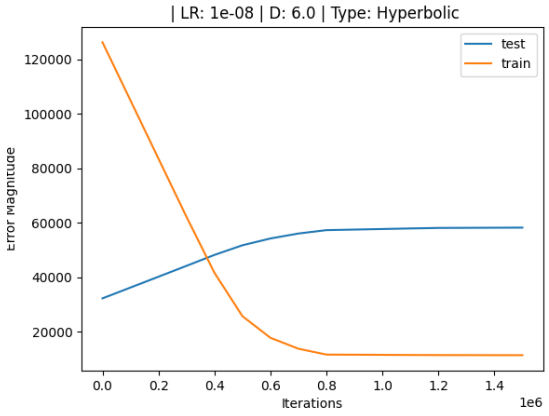


Figure 1.40: Degree 6

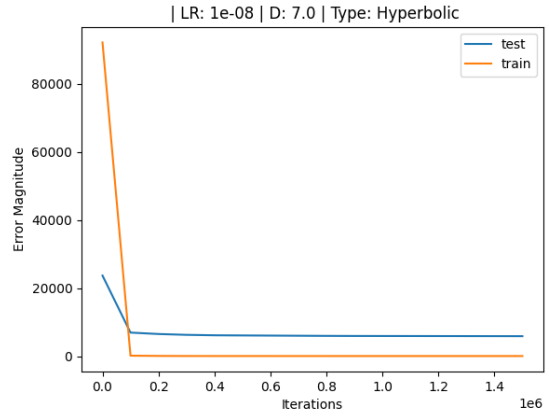


Figure 1.41: Degree 7

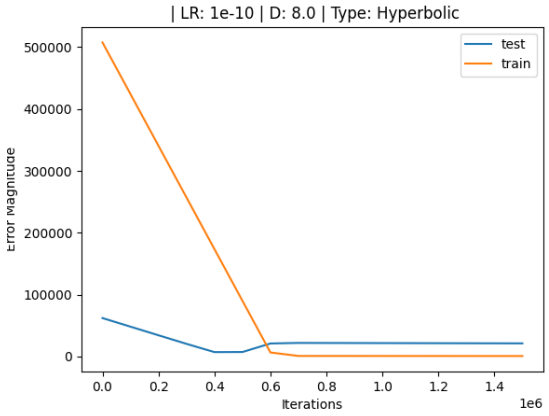


Figure 1.42: Degree 8

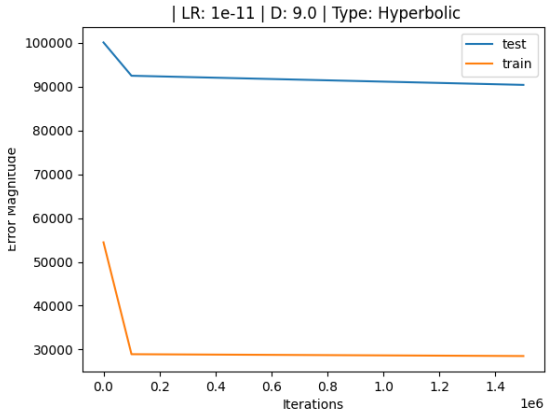


Figure 1.43: Degree 9

Interpretation: Graph 1

- The training error decreases as we increase the degree till the degree 7, hence as we increase the complexity of the model the training error decreases
- The test error is really high for degrees 1 to 3, hence it generalizes poorly. After this the test error falls considerably.
- Both the error curves achieve a minima at 7 after which the error curves start increasing, pointing out that 7 is the best degree.

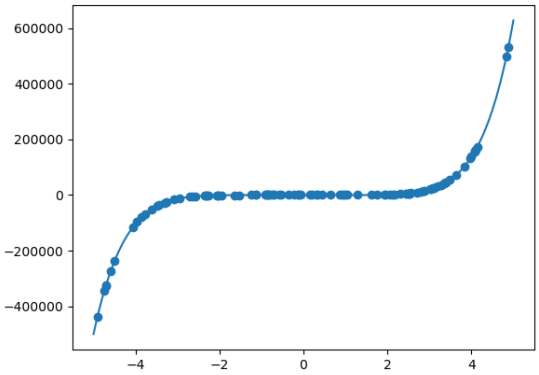
Interpretation: Graphs 2-10

- As we can see for degree 1,2 and 3 the testing error is increasing and hasn't yet flattened, this signifies that it might still increase if we increase the number of iterations.
- In degree 5 we see a peculiar drop in test error initially after which it starts increasing continuously,
- For all the degrees we see that the training error has a high value after which it drops suddenly.
- For Degree 9 both the curves flatten out but still there is a considerable difference between the magnitude of both the errors.

Conclusion:
For every error function we get seven as the best degree. Hence degree seven can be termed as the best degree regardless of the error function. So again we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 2000000
Training Error: 15.093
Testing Error: 188.422
Estimated Noise variance: 344.359
mean difference: -11.722
Degree: 7
Learning Rate: 10^{-7}

Figure 1.44: The estimated polynomial

Estimated Polynomial $4.96 + 3.10x + 1.75x^2 + 1.54x^3 + 1.94x^4 + 5.48x^5 + 4.01x^6 + 6.99x^7$

1.4 With Regularization - trained on 20 data points

This section deals with the effects of regularization on our learning model. First we identify the degrees for which we faced over-fitting in the previous sections so that we can apply regularization to improve the result and reduce the testing error. Then we will find the optimal regularization parameter for each of those degrees using graphs. Once we have the optimal regularization parameter we will again plot error vs iterations graph for those degrees. Then these graphs are used to interpret what was the effect of regularization.

1.4.1 Log sigmoid Error Function

For this error function we observe from section 1.3 above that we face the problem of over-fitting for polynomials of degree 2,4,5 and 6. Hence we add regularization only to these degrees so as to reduce the testing error. We also use regularization for degree 7 as it the polynomial which fits the best, as is observed above, to check if we can obtain any better results for it using regularisation.

Degree	Learning Rate	Regularisation Parameter	Training Error	Test Error
2	10^{-5}	100	3.07 10^{9}	3.98 10^{10}
4	10^{-6}	2000	6.88 10^{8}	7.56 10^{10}
5	10^{-7}	40000	3.92 10^{8}	4.79 10^{9}
6	10^{-8}	700000	1.14 10^{8}	4.28 10^{10}

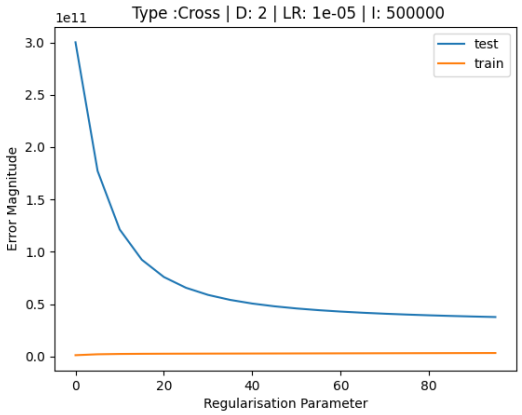


Figure 1.45: Degree 2

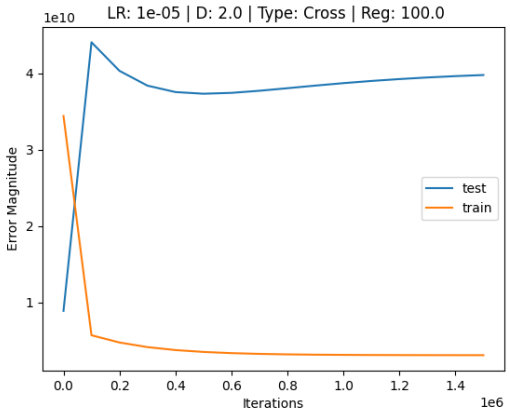


Figure 1.46: Degree 2

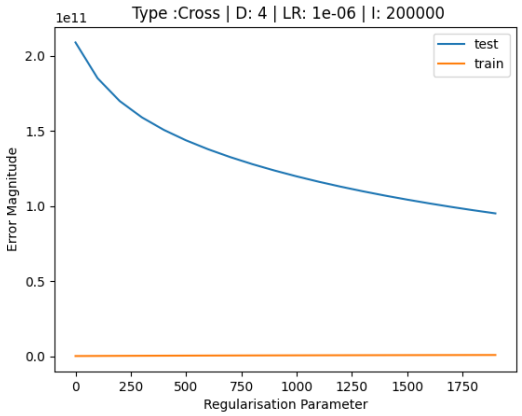


Figure 1.47: Degree 4

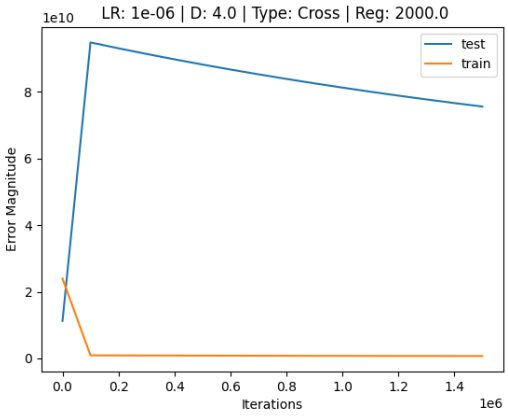


Figure 1.48: Degree 4

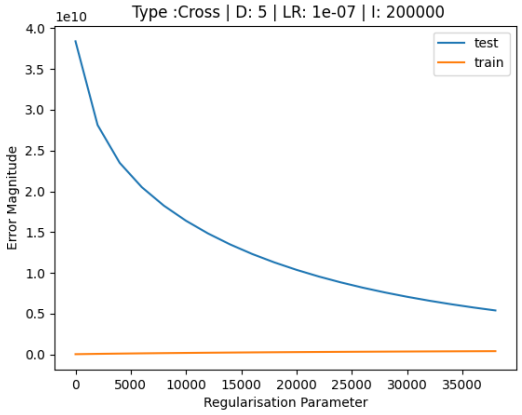


Figure 1.49: Degree 5

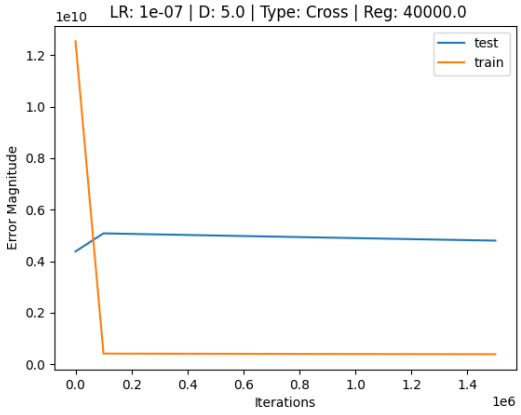


Figure 1.50: Degree 5

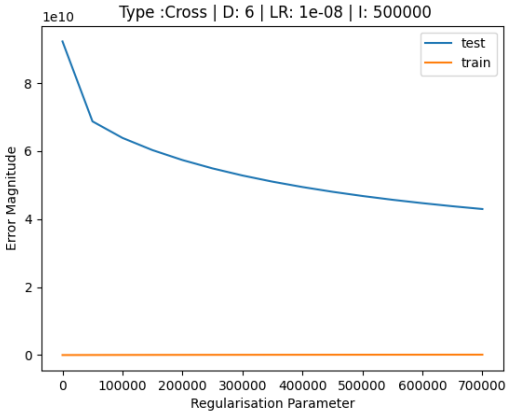


Figure 1.51: Degree 6

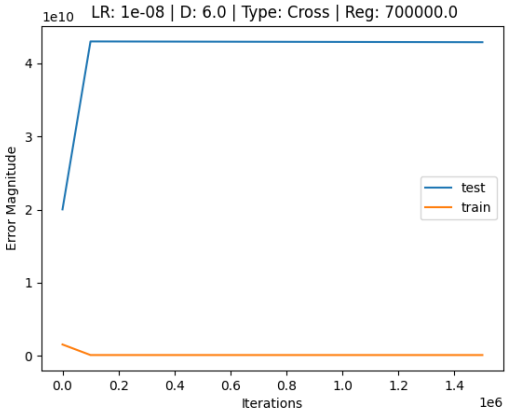


Figure 1.52: Degree 6

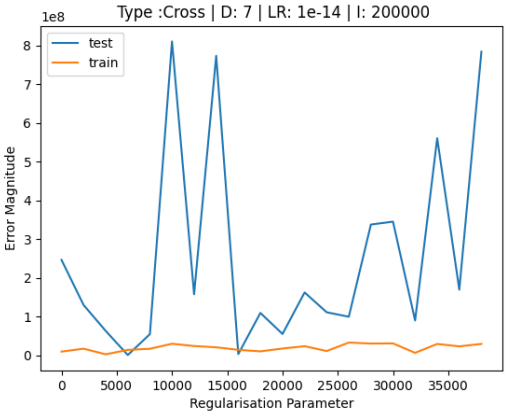


Figure 1.53: Degree 7

Interpretation:

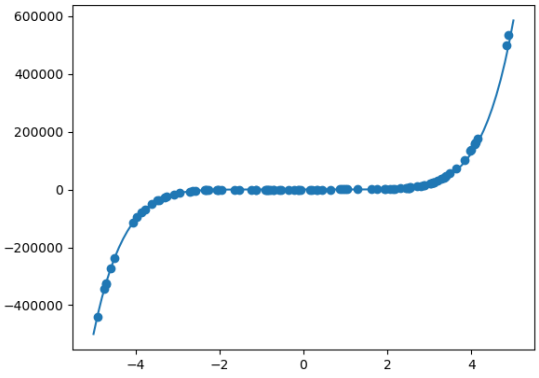
- We see a clear drop in the testing set error for degree 2,4,5,6 as the regularisation parameter increases.
- This suggests that the polynomials indeed suffered from overfitting on the training data set and regularisation helped to solve the issue.
- On comparing the results with the results obtained without regularisation, we observe that for degree 4 and 5 there is a decrease in the test error by a factor of 10, although this occurs at the cost of a slight increase in the training error.
- For degree 7 we observe a very inconclusive graph. This is explained by the fact the graph for degree 7 without regularisation had both the testing and training error decreasing and converging almost at the same point which implies that there was no problem of overfitting.

Conclusion:
Even after regularization the error magnitude of degree 7 remains the lowest. Hence it is our choice for the degree.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.

After checking for various regularization parameter we could not find a suitable parameter for degree 7, as all of them had error greater than the with no parameter.



Number of Iterations: 1000000
Training Error: 1060.2778021
Testing Error: 53729436.08059
Estimated Noise variance: 1060.276
mean difference: 11.428
Degree: 7
Learning Rate: 10^{-9}
regularization parameter: 0

Figure 1.54: The estimated polynomial

Polynomial Guessed: $4.96 + 4.09x - 0.23x^2 + 4.56x^3 + 1.80x^4 + 2.04x^5 + 2.63x^6 + 6.84x^7$

1.4.2 Hyperbolic Error Function

For this error function we observe from section 1.3 above that we face the problem of overfitting for polynomials of degree 1,2 and 3. Hence we add regularization only to these degrees so as to reduce the testing error. We also use regularization for degree 7 as it the polynomial which fits the best, as is observed above, to check if we can obtain any better results for it using regularisation.

Degree	Learning Rate	Regularisation Parameter	Training Error	Test Error
1	1	0.001	$9.99\ 10^4$	$3.48\ 10^4$
2	0.1	0.01	$8.38\ 10^4$	$6.40\ 10^4$
3	0.01	0.1	$6.79\ 10^4$	$2.49\ 10^4$
7	10^{-8}	0.1	31.57	$2.66\ 10^3$

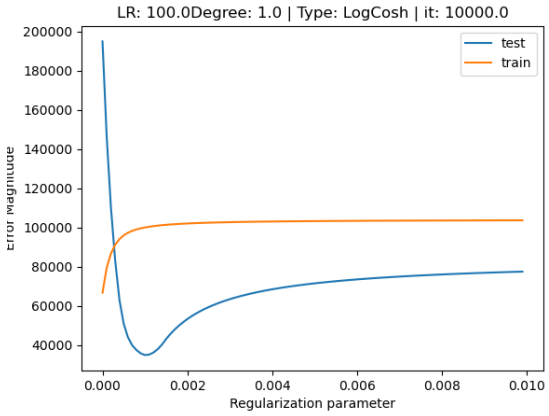


Figure 1.55: Degree 1

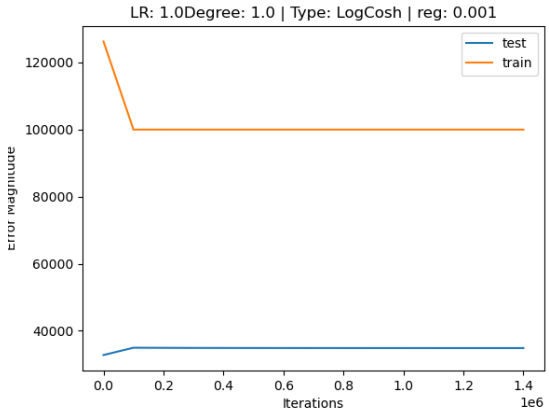


Figure 1.56: Degree 1

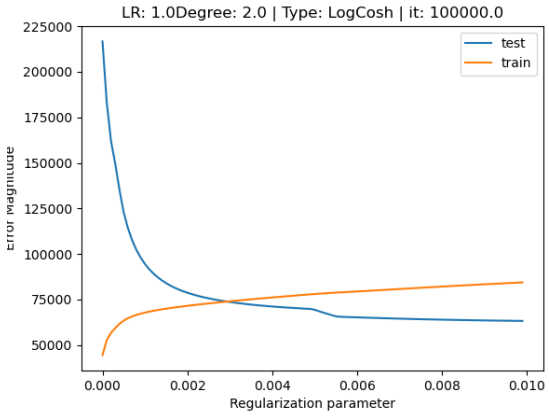


Figure 1.57: Degree 2

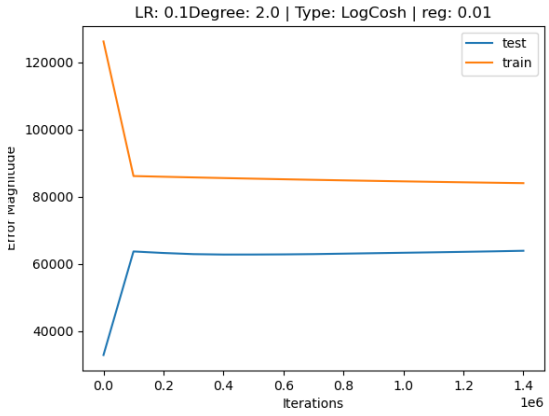


Figure 1.58: Degree 2

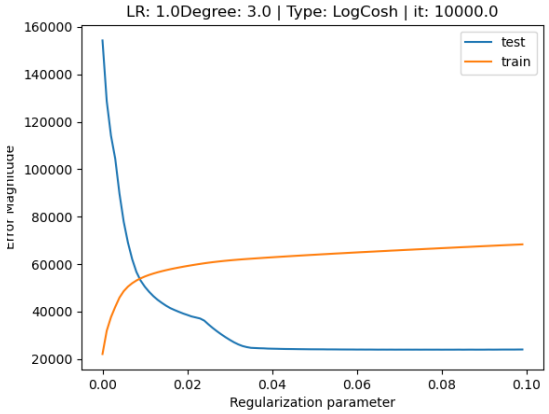


Figure 1.59: Degree 3

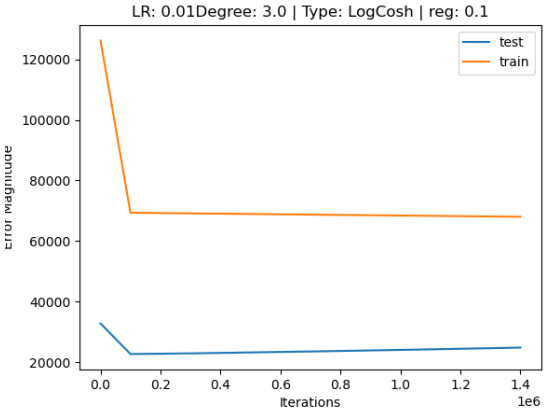


Figure 1.60: Degree 3

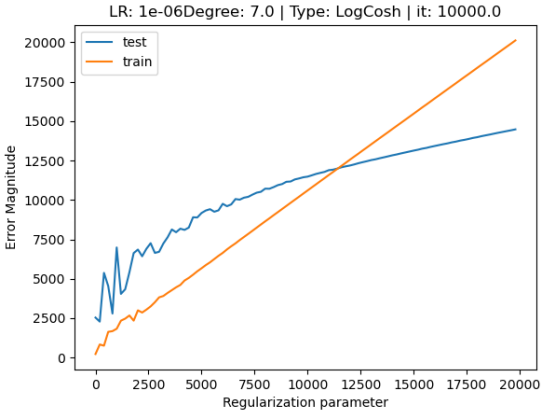


Figure 1.61: Degree 7

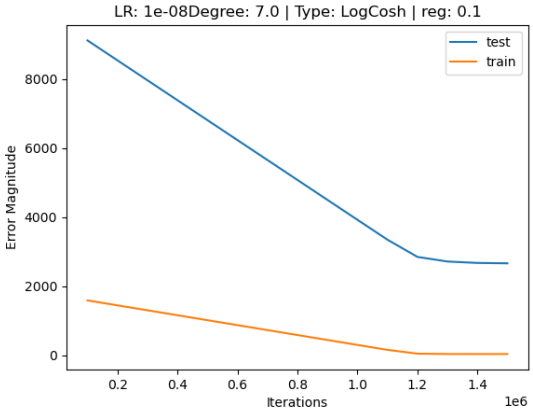


Figure 1.62: Degree 7

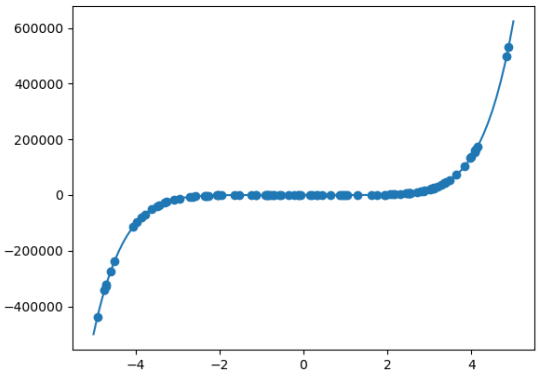
Interpretation:

- As we can see for degree 1,2 and 3 the testing error starts decreasing considerably on introducing regularisation although this increases the training error slightly.
- We observe that for degrees 1,2,3 there is a decrease in the testing error by a factor of 10 after adding regularisation with a very little rise in the training error which clearly implies that regularisation has helped.
- We also observe that the training error decreases a lot for degree 7.
- As can be observed from the graph without regularisation for degree 7 that there isn't any problem of overfitting, hence the training and testing errors keep increasing on increasing regularisation parameter as can be seen from figure 1.62
- From the figure 1.56 we observe a clear minima in the testing error for degree 1 suggesting an optimum value of regularisation parameter as 0.001

Conclusion:
From the above interpretations we can safely conclude that degree seven is the best fit for this error function since it still offers the least training and testing error inspite of adding regularsiation to degrees 1,2,3. So now we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value.



Number of Iterations: 2000000
Degree: 7
Learning Rate: 10^{-7}
Training Error: 13.10
Testing Error: 427.35
Estimated Noise variance: 267.845
Mean of error: -12.28
Regularization parameter: 0.1

Figure 1.63: The estimated polynomial

Estimated Polynomial $0.95 + 0.11x + 1.71x^2 + 3.59x^3 + 1.96x^4 + 5.01x^5 + 3.89x^6 + 6.98x^7$

1.4.3 Square Root Error Function

For this error function we observe from section 1.3 above that we face the problem of over-fitting for polynomials of degree 2,4,5 and 6. Hence we add regularization only to these degrees so as to reduce the testing error. We also use regularization for degree 7 as it the polynomial which fits the best, as is observed above, to check if we can obtain any better results for it using regularisation.

Degree	Learning Rate	Regularisation Parameter	Training Error	Test Error
2	10^{-5}	100	3.07 10^{9}	3.98 10^{10}
4	10^{-6}	2000	6.88 10^{8}	7.56 10^{10}
5	10^{-7}	40000	3.92 10^{8}	4.79 10^{9}
6	10^{-8}	700000	1.14 10^{8}	4.28 10^{10}

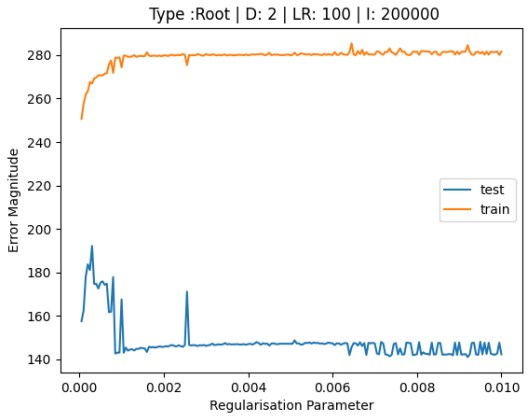


Figure 1.64: Degree 2

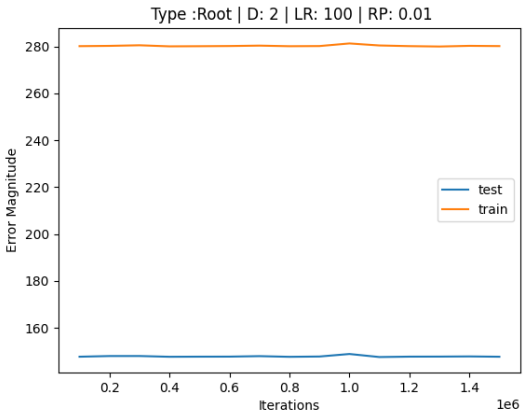


Figure 1.65: Degree 2

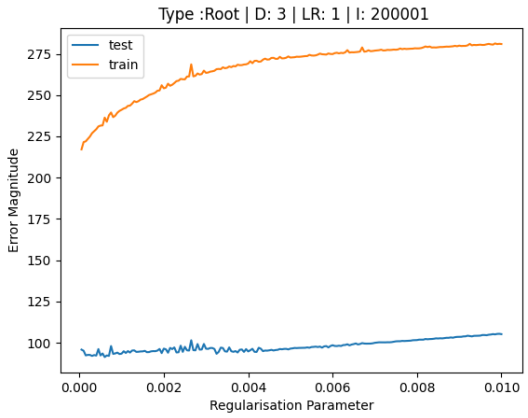


Figure 1.66: Degree 3

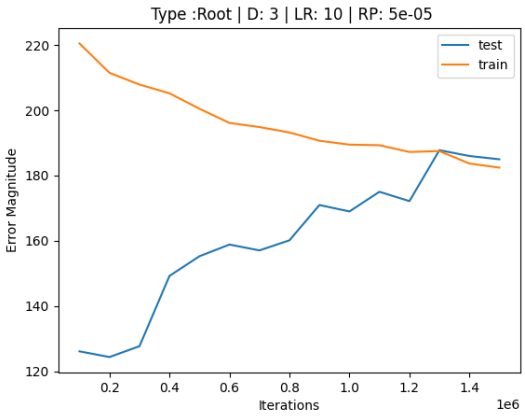


Figure 1.67: Degree 3

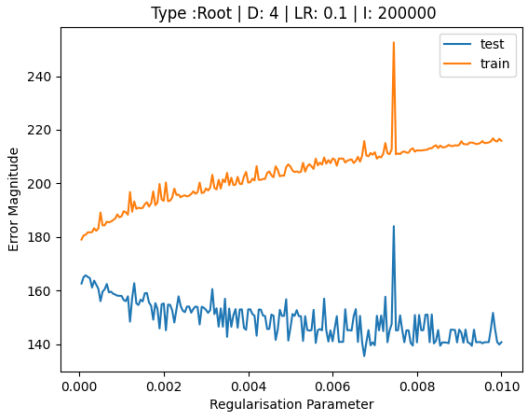


Figure 1.68: Degree 4

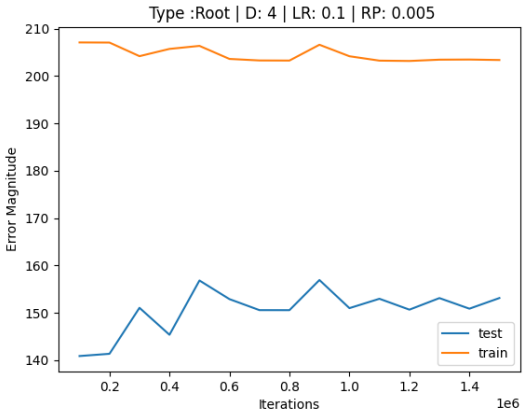


Figure 1.69: Degree 4

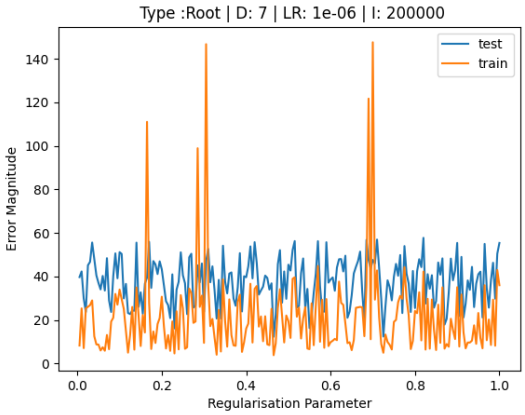


Figure 1.70: Degree 7

Interpretation:

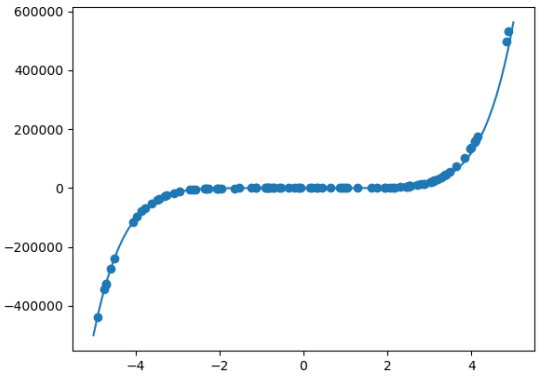
- We see a clear drop in the testing set error for degree 2,4,5,6 as the regularisation parameter increases.
- This suggests that the polynomials indeed suffered from overfitting on the training data set and regularisation helped to solve the issue.
- On comparing the results with the results obtained without regularisation, we observe that for degree 4 and 5 there is a decrease in the test error by a factor of 10, although this occurs at the cost of a slight increase in the training error.
- For degree 7 we observe a very inconclusive graph. This is explained by the fact the graph for degree 7 without regularisation had both the testing and training error decreasing and converging almost at the same point which implies that there was no problem of overfitting.

Conclusion:
Even after regularization the error magnitude of degree 7 remains the lowest. Hence it is our choice for the degree.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.

After checking for various regularization parameter we could not find a suitable parameter for degree 7, as all of them had error greater than with no parameter.



Number of Iterations: 1500000
Training Error: 5.83593
Testing Error: 38.22695
Estimated Noise variance: 2546.56184
mean difference: 10.1185473
Degree: 7
Learning Rate: 10^{-6}

Figure 1.71: The estimated polynomial

Estimated Polynomial $2.97 + 1.07x + 0.84x^2 + 4.33x^3 + 0.43x^4 + 0.42x^5 + 2.01x^6 + 6.77x^7$

1.4.4 Mean Absolute Error Function

For this error function we observe from section 1.3 above that we face the problem of over-fitting for polynomials of degree 4,5 and 6. Hence we add regularization only to these degrees so as to reduce the testing error. We also use regularization for degree 7 as it the polynomial which fits the best, as is observed above, to check if we can obtain any better results for it using regularisation.

Degree	Learning Rate	Regularisation Parameter	Training Error	Test Error
4	10^{-4}	100	$3.72\ 10^4$	$1.19\ 10^5$
5	10^{-4}	10	$9.95\ 10^3$	$2.79\ 10^4$
6	10^{-5}	100	$6.74\ 10^5$	$1.25\ 10^5$

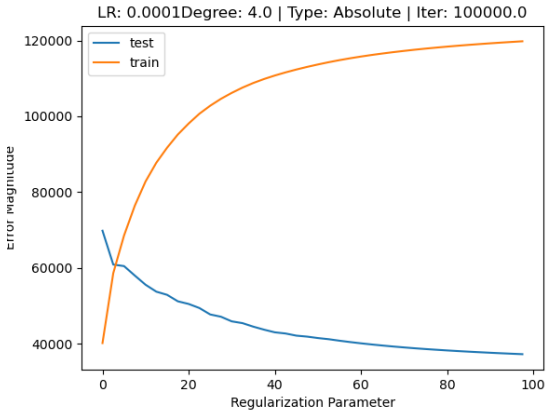


Figure 1.72: Degree 4

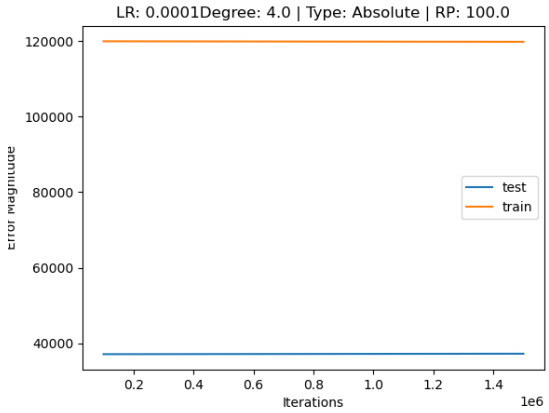


Figure 1.73: Degree 4

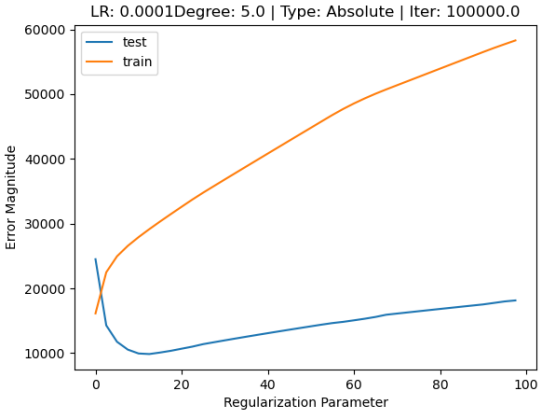


Figure 1.74: Degree 5

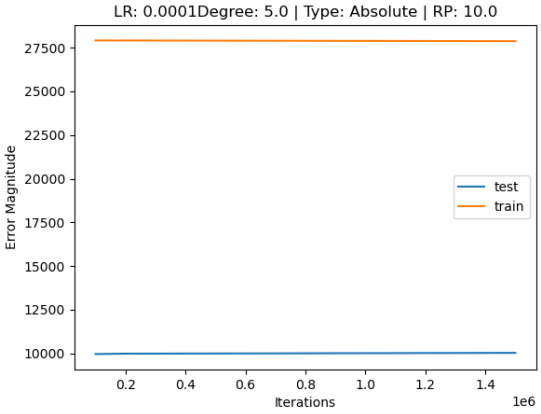


Figure 1.75: Degree 5

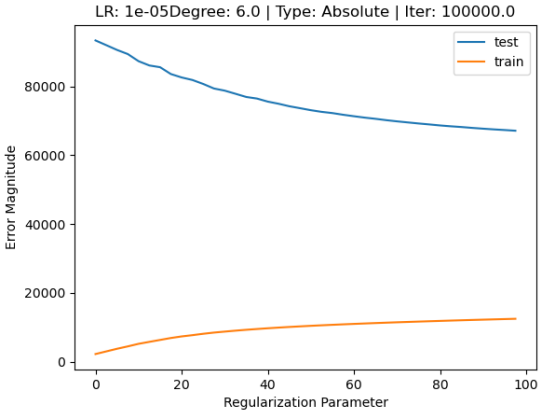


Figure 1.76: Degree 6

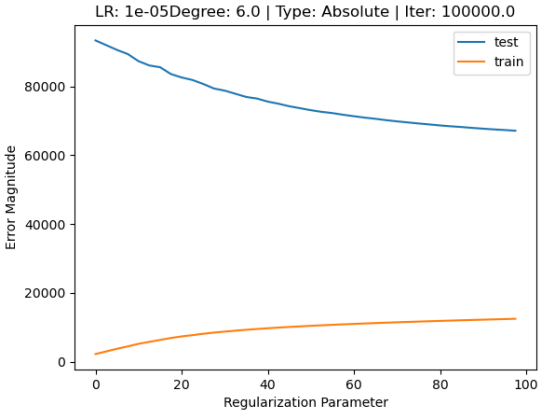


Figure 1.77: Degree 6

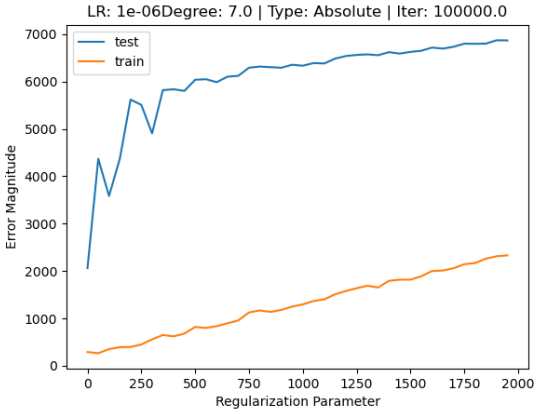


Figure 1.78: Degree 7

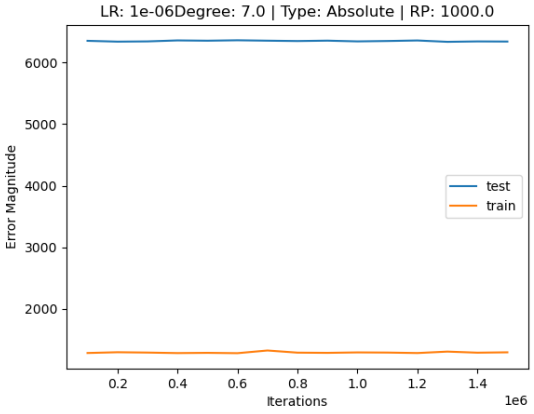


Figure 1.79: Degree 7

Interpretation:

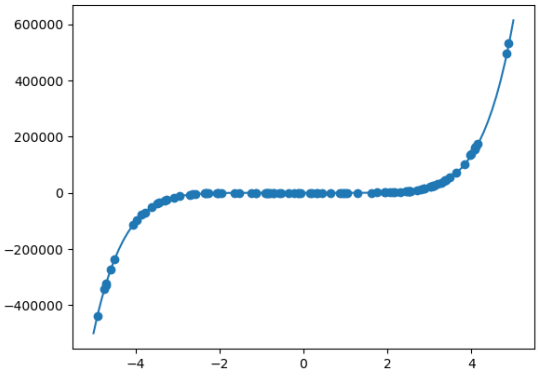
- We see a clear drop in the testing set error for degree 2,4,5,6 as the regularisation parameter increases.
- This suggests that the polynomials indeed suffered from overfitting on the training data set and regularisation helped to solve the issue.
- On comparing the results with the results obtained without regularisation, we observe that for degree 4 and 5 there is a decrease in the test error by a factor of 10, although this occurs at the cost of a slight increase in the training error.
- For degree 7 we observe a very inconclusive graph. This is explained by the fact the graph for degree 7 without regularisation had both the testing and training error decreasing and converging almost at the same point which implies that there was no problem of overfitting.

Conclusion:
Even after regularization the error magnitude of degree 7 remains the lowest. Hence it is our choice for the degree.-

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.

After checking for various regularization parameter we could not find a suitable parameter for degree 7, as all of them had error greater than with no parameter.



Number of Iterations: 1400000
Degree: 7
Learning Rate: 10^{-7}
Training Error: 19.448532260627598
Testing Error: 2526.9021938444153
Estimated Noise variance: 472.524
Mean of error: -1.9442

Figure 1.80: The estimated polynomial

Estimated Polynomial $-4.38 * 10^{-2} + 2.11x + 0.71x^2 + 4.68x^3 + 0.61x^4 + 2.18x^5 + 2.88x^6 + 6.88x^7$

1.5 No regularization - 90:10 training:testing data split

1.5.1 Mean absolute Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{-1}	$3.86\ 10^4$	$1.04\ 10^5$
2	10^{-2}	$3.86\ 10^4$	$1.12\ 10^5$
3	10^{-3}	$2.47\ 10^4$	$5.87\ 10^4$
4	10^{-4}	$2.60\ 10^4$	$8.19\ 10^4$
5	10^{-5}	$1.01\ 10^4$	$1.67\ 10^4$
6	10^{-6}	$1.07\ 10^4$	$1.98\ 10^4$
7	10^{-8}	$4.08\ 10^2$	$6.15\ 10^2$
8	10^{-10}	$1.71\ 10^4$	$1.21\ 10^5$
9	10^{-10}	$3.36\ 10^3$	$3.61\ 10^3$

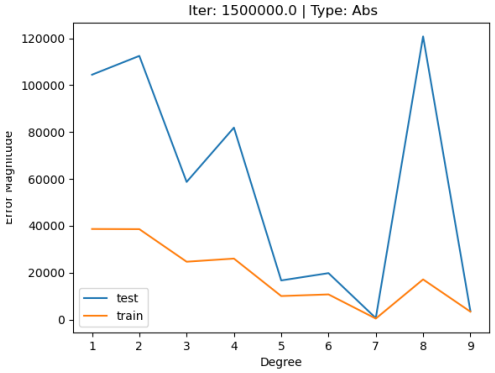


Figure 1.81: 1500000 iterations

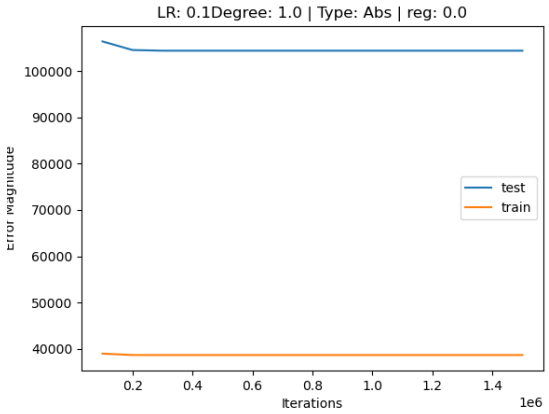


Figure 1.82: Degree 1

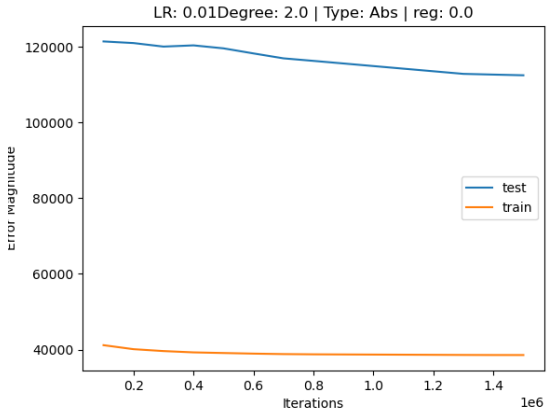


Figure 1.83: Degree 2

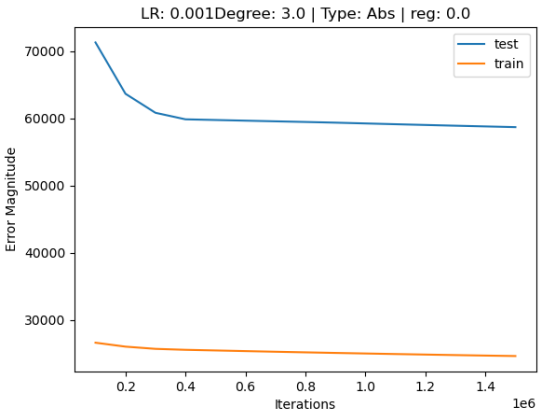


Figure 1.84: Degree 3

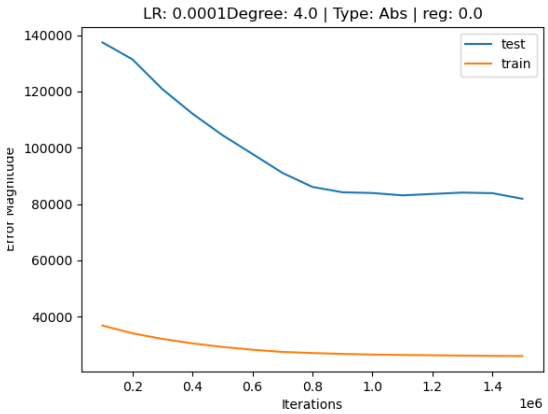


Figure 1.85: Degree 4

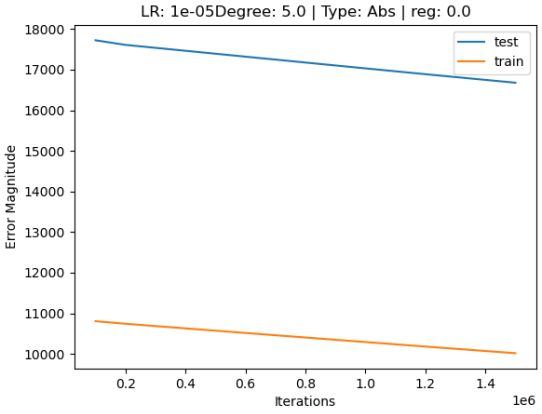


Figure 1.86: Degree 5

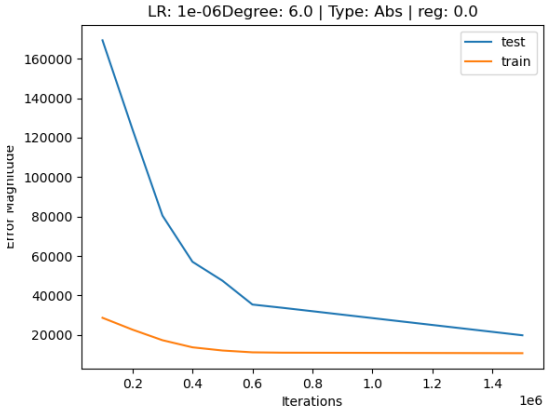


Figure 1.87: Degree 6

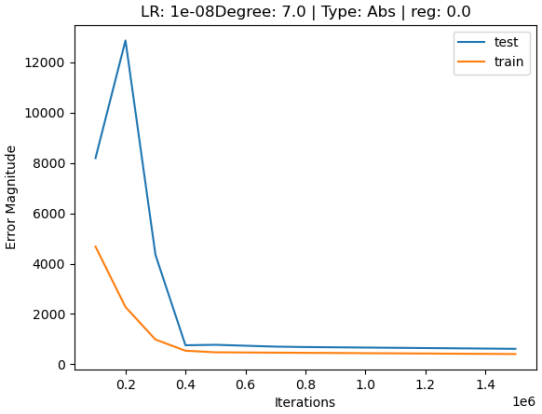


Figure 1.88: Degree 7

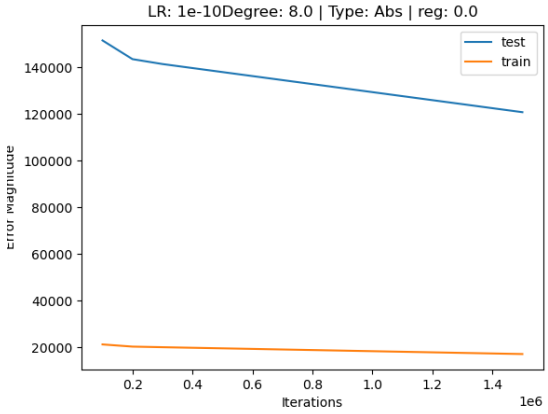


Figure 1.89: Degree 8

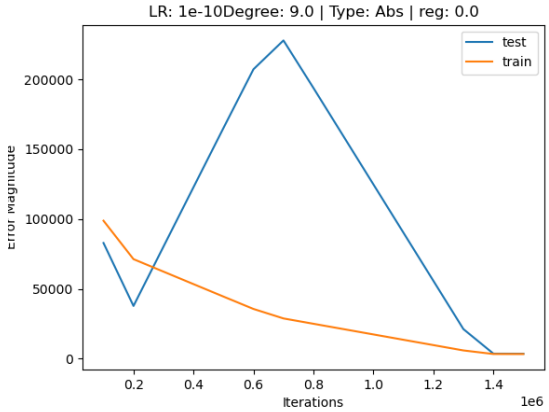


Figure 1.90: Degree 9

Interpretation: Graph 1

- The training error and testing error both decrease as we increase the degree till 7 implying that on increasing the complexity of the model we get a better fit on the data.
- The testing error shoots up for degree 8 giving a clear evidence of over fit on the training data.
- We again obtain the least errors for degree 7 and hence 7 is our guess for the degree of the underlying polynomial.
- On comparing this result with the one obtained on training on 20 data points, we observe a decrease in the testing error for degree 7. This is because the training data was much smaller earlier and hence the model could not fit the entire data set nicely.

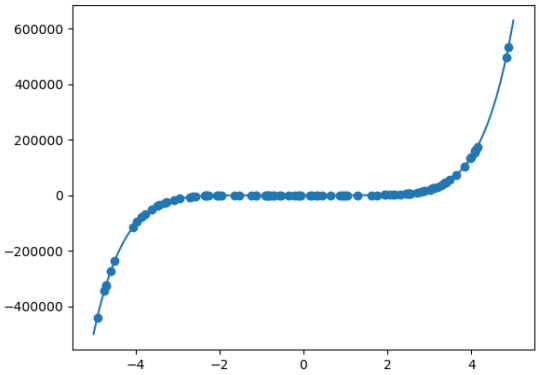
Interpretation: Graphs 2-10

- The graph for degree 1,2 and 3 is almost flat throughout. This is because the model has already converged for a very small number of iterations and hence there is not much difference in the error after that.
- From graphs for degree 1,2,3,5 we observe that the testing error is much larger than the training error. This could be because of over fitting on the training data.
- For degree 7 and 9 there is an initial increase in the testing error. This could be because of an initially the weights are moving away from the minima due to a large learning rate, after which they start converging towards the minima and hence the testing error decreases.

Conclusion:
From the above interpretations we conclude that degree seven is the best fit for this error function. So now we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1000000
Degree: 7
Learning Rate: 10^{-7}
Training Error: 13.401
Testing Error: 24.228
Estimated Noise variance: 302.255
Mean of error: -0.274

Figure 1.91: The estimated polynomial

Estimated Polynomial $3.99 + 2.12x + 2.11x^2 + 2.68x^3 + 3.77x^4 + 5.98x^5 + 3.99x^6 + 6.98x^7$

1.5.2 Log sigmoid Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{-1}	$7.90\ 10^{9}$	$2.28\ 10^{9}$
2	10^{-4}	$5.91\ 10^{9}$	$2.12\ 10^{10}$
3	10^{-4}	$4.93\ 10^{8}$	$9.66\ 10^{9}$
4	10^{-6}	$1.02\ 10^{9}$	$7.71\ 10^{8}$
5	10^{-6}	$3.40\ 10^{7}$	$2.39\ 10^{7}$
6	10^{-8}	$5.46\ 10^{7}$	$2.13\ 10^{8}$
7	10^{-9}	$2.05\ 10^{2}$	$1.60\ 10^{2}$
8	10^{-10}	$2.24\ 10^{3}$	$3.56\ 10^{3}$
9	10^{-13}	$1.77\ 10^{7}$	$4.02\ 10^{7}$

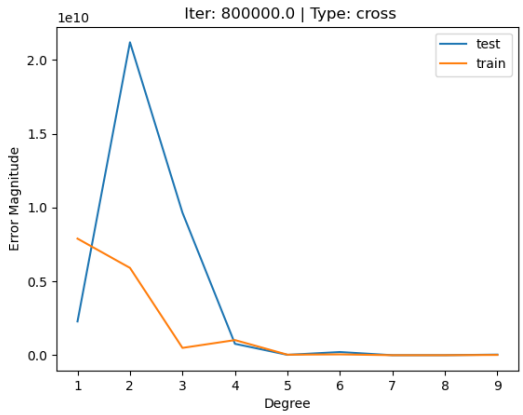


Figure 1.92: 800000 iterations

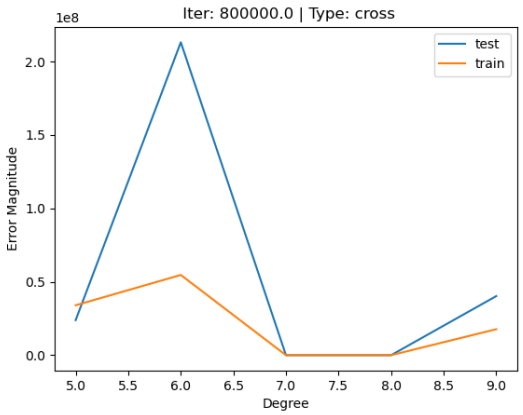


Figure 1.93: 800000 iterations

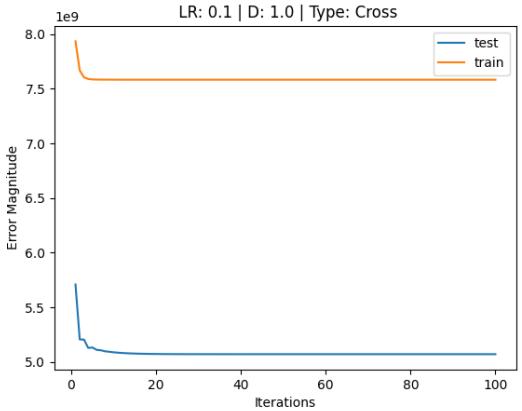


Figure 1.94: Degree 1

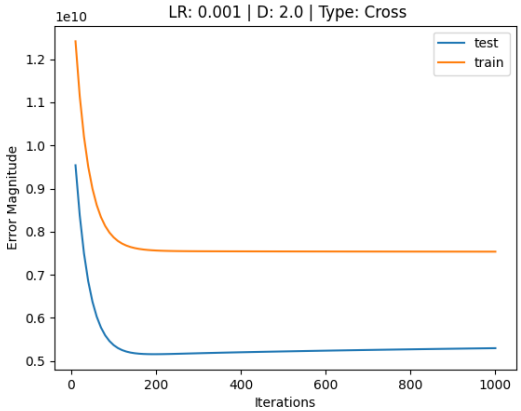


Figure 1.95: Degree 2

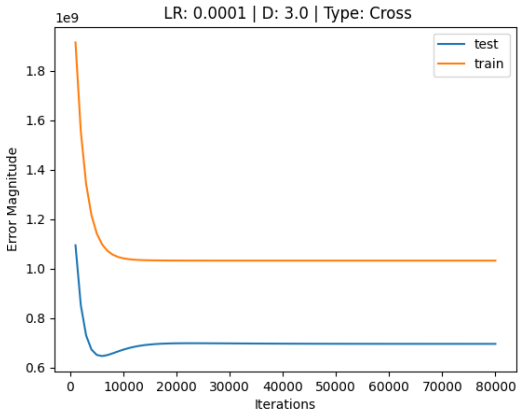


Figure 1.96: Degree 3

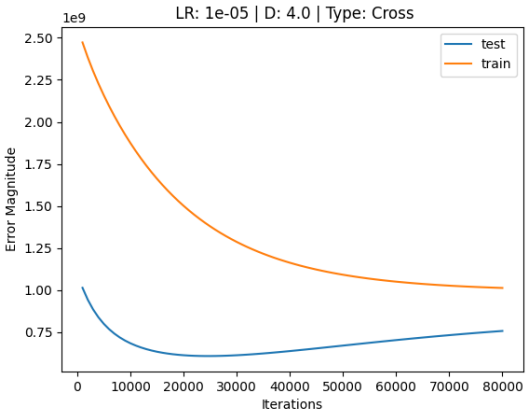


Figure 1.97: Degree 4

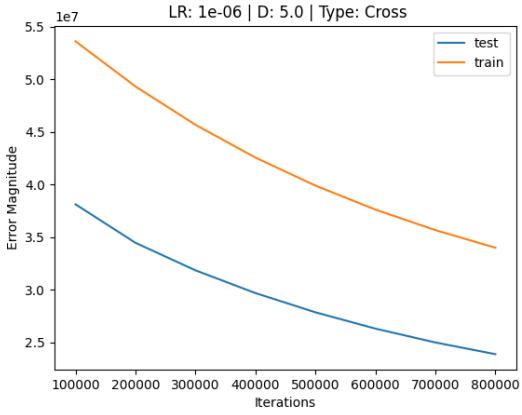


Figure 1.98: Degree 5

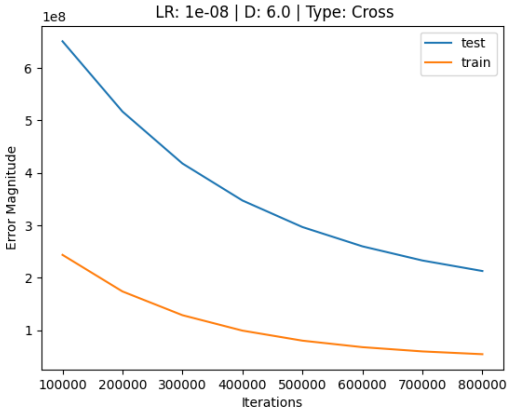


Figure 1.99: Degree 6

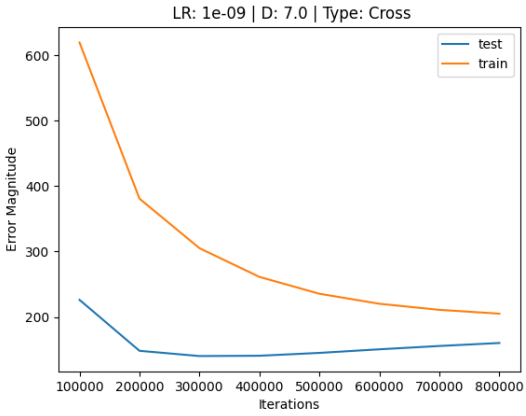


Figure 1.100: Degree 7

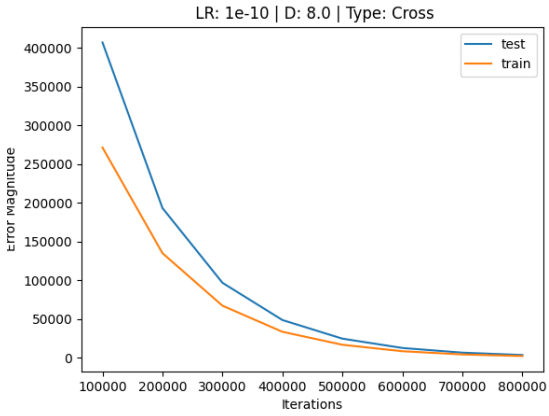


Figure 1.101: Degree 8

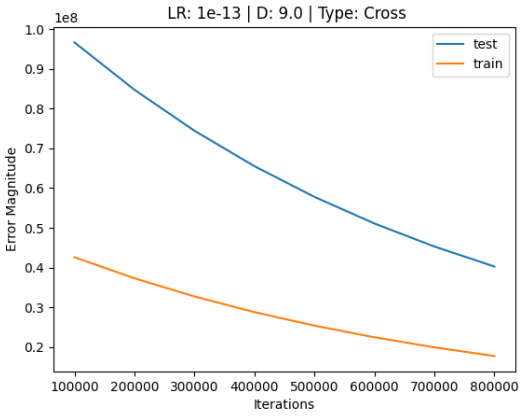


Figure 1.102: Degree 9

Interpretation: Graph 1 and 2

- We observe relatively very high errors for degree 1,2,3,4 as compared to the other degrees. Hence, we chose to make 2 separate graphs to have a better understanding of the changes in the error for higher degrees.
- We once again observe a huge spike for degree 6 suggesting a very poor fit on the entire data set.
- We observe a similar spike for degree 2.
- The training and testing error almost flattens out for degree 7 and 8 after which it increases for degree.
- This implies that even degree 8 was a good fit on the data this time, but since degree 7 has a lower training and testing error, we again choose degree 7 as out guess.

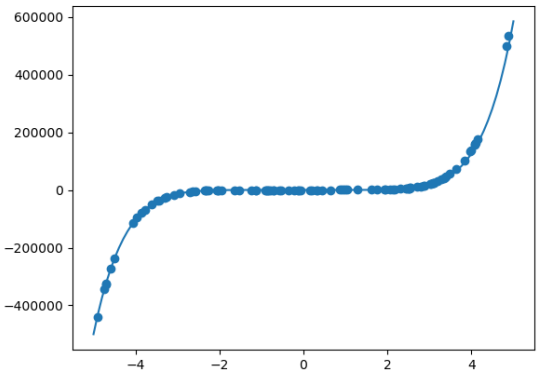
Interpretation: Graphs 3-11

- We observe that for degree 1,2 and 3 the model converges in very less number of iterations as compared to other degrees.
- The higher training error for degree 1 suggests an under fit on the training data.
- Although the graph for degree 9(Fig 1.85) shows a considerable difference between the training and testing error but the shape of the graph suggests that if the number of iterations were to be increased the testing error would have decreased further i.e. the model has not converged yet.
- There is a huge difference in the errors of degree 7 and other degrees and hence degree 7 is the best polynomial for this data.

Conclusion:
Again we are getting degree 7 to be the best guess for the polynomial as can be seen from interpretations of graph 1 and graphs 2 - 10. We get a difference of a factor of 10 in the error for degree 7 and the next best error i.e. degree 8.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1000000
Training Error: 205.193
Testing Error: 160.379
Estimated Noise variance: 205.138
mean difference: -3.342
Degree: 7
Learning Rate: 10^{-10}

Figure 1.103: The estimated polynomial

Polynomial Guessed: $3.01 + 3.06x + 1.12x^2 + 3.54x^3 + 3.90x^4 + 5.84x^5 + 3.99x^6 + 6.987x^7$

1.5.3 Mean Root Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{4}	$1.38\ 10^{2}$	$1.51\ 10^{2}$
2	10^{1}	$1.43\ 10^{2}$	$1.02\ 10^{2}$
3	10^{1}	$1.12\ 10^{2}$	$6.99\ 10^{1}$
4	10^{-1}	$1.12\ 10^{2}$	$1.19\ 10^{2}$
5	10^{-2}	$5.72\ 10^{1}$	$8.67\ 10^{1}$
6	10^{-2}	$5.96\ 10^{1}$	$8.40\ 10^{1}$
7	10^{-5}	$2.83\ 10^{0}$	$3.01\ 10^{0}$
8	10^{-6}	$1.51\ 10^{1}$	$2.13\ 10^{1}$
9	10^{-7}	$2.55\ 10^{1}$	$2.53\ 10^{1}$

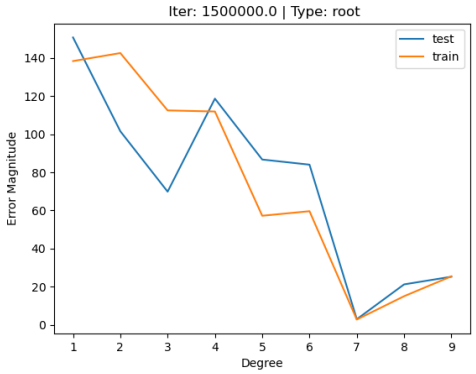


Figure 1.104: 1500000 iterations

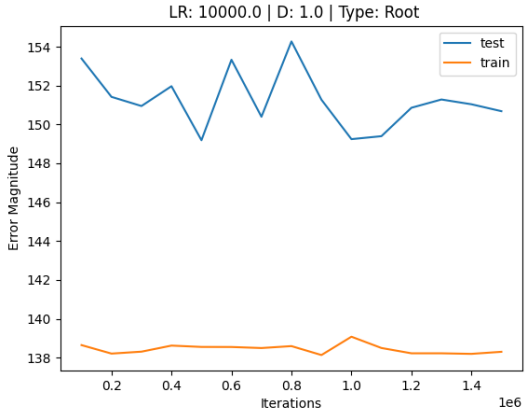


Figure 1.105: Degree 1

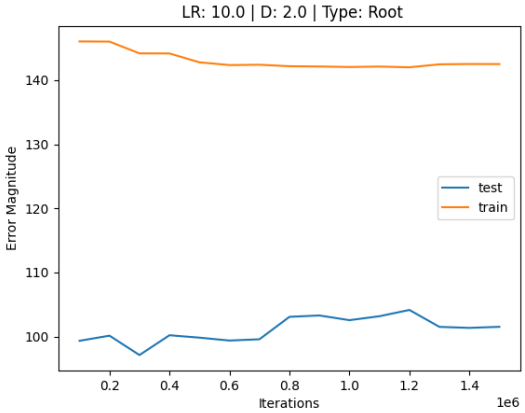


Figure 1.106: Degree 2

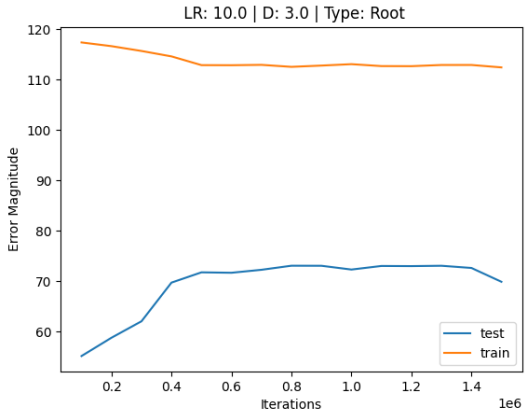


Figure 1.107: Degree 3

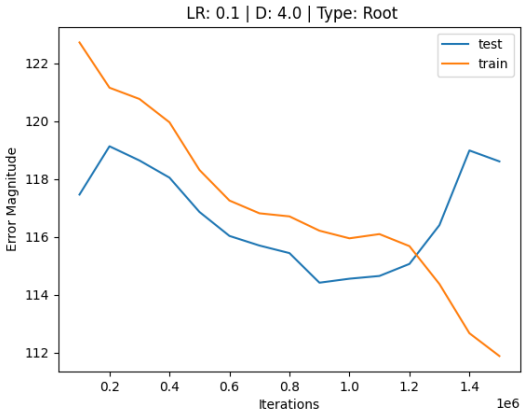


Figure 1.108: Degree 4

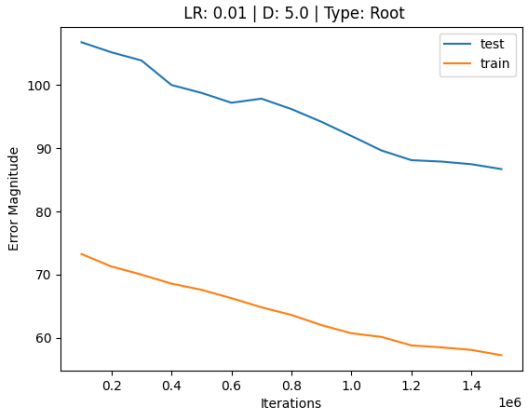


Figure 1.109: Degree 5

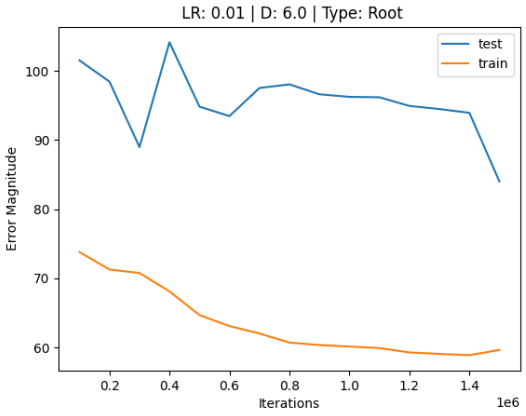


Figure 1.110: Degree 6

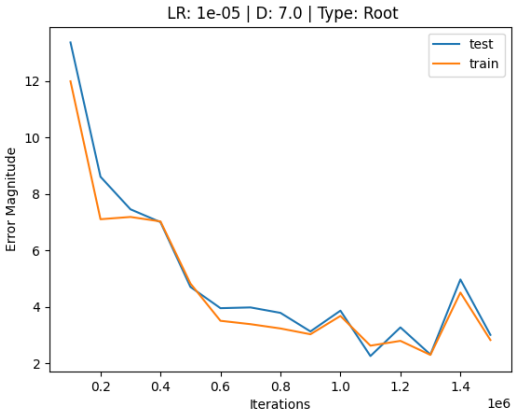


Figure 1.111: Degree 7

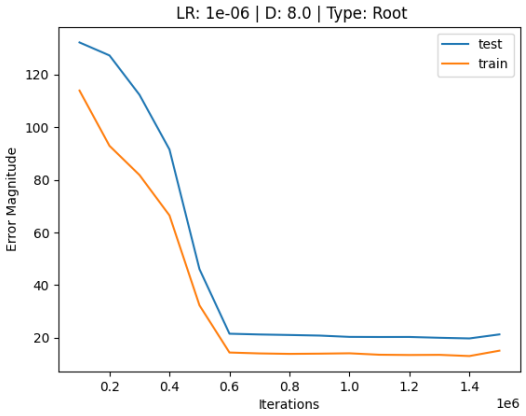


Figure 1.112: Degree 8

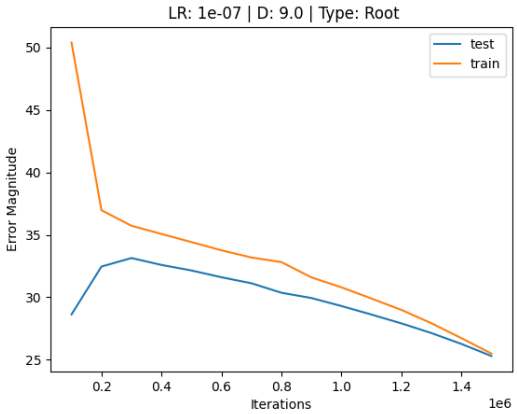


Figure 1.113: Degree 9

Interpretation: Graph 1

- The training and testing error again decrease on increasing the degree till degree 7.
- We clearly observe the lowest training and testing errors for degree 7.
- The testing set error decreases by a factor of 10 for degree 7 when compared to the case of training on 20 data points.
- Since the testing error increases for degree 8 and 9, there is a possibility of over fit on the training data for these degrees. Hence, degree 7 seems to be the best polynomial fit on the data.

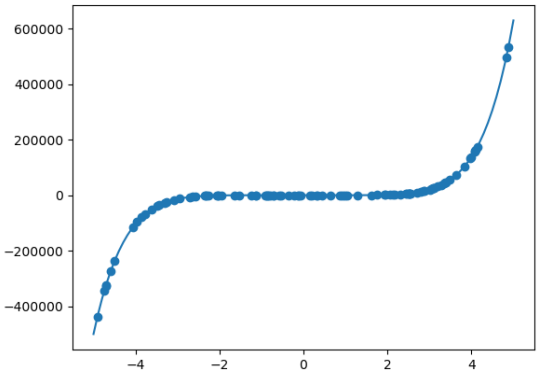
Interpretation: Graphs 2-10

- The large difference in the training and testing errors for degree 1,5,6 suggest that these models are an over fit on the training data.
- The higher testing error for degree 2 and 3 imply that these models don't fit the data very well and hence are an under fit.
- For degree 4 we observe that the testing error starts increasing after around 8,00,000 iterations which could be because the model is moving away from the minima i.e diverging.
- Similarly, we observe a spike near the end for degree 7 which could mean the same thing as above(diverging).
- The graph for degree 8 becomes flat after 6,00,000 iterations which means the model has already converged.

Conclusion:
Again degree 7 seems to be the best guess. So now we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 1500000
Training Error: 2.829
Testing Error: 3.011
Estimated Noise variance: 943.900
mean difference: -5.754
Degree: 7
Learning Rate: 10^{-5}

Figure 1.114: The estimated polynomial

Estimated Polynomial $3.16 + 1.58x + 0.45x^2 + 6.50x^3 + 4.21x^4 + 5.55x^5 + 3.97x^6 + 6.99x^7$

1.5.4 Hyperbolic Error Function

Degree	Learning Rate	Training Error	Test Error
1	10^{1}	$4.45\ 10^{4}$	$5.36\ 10^{4}$
2	10^{0}	$4.44\ 10^{4}$	$5.28\ 10^{4}$
3	10^{-2}	$2.23\ 10^{4}$	$2.99\ 10^{4}$
4	10^{-4}	$2.83\ 10^{4}$	$3.80\ 10^{4}$
5	10^{-5}	$1.06\ 10^{4}$	$1.12\ 10^{4}$
6	10^{-8}	$4.93\ 10^{4}$	$4.78\ 10^{4}$
7	10^{-8}	$3.16\ 10^{2}$	$3.48\ 10^{2}$
8	10^{-10}	$1.10\ 10^{4}$	$2.07\ 10^{4}$
9	10^{-10}	$5.63\ 10^{3}$	$4.12\ 10^{3}$

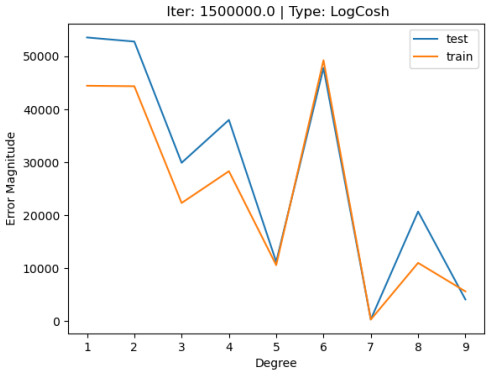


Figure 1.115: 1500000 iterations

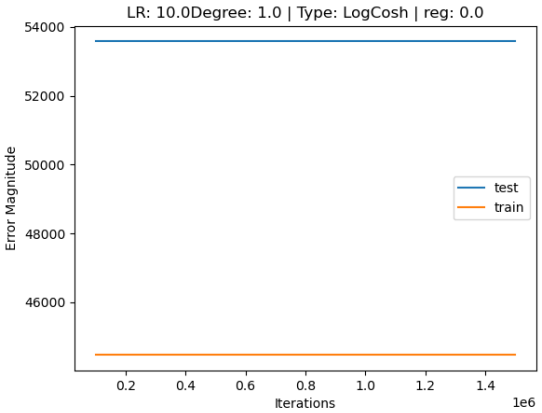


Figure 1.116: Degree 1

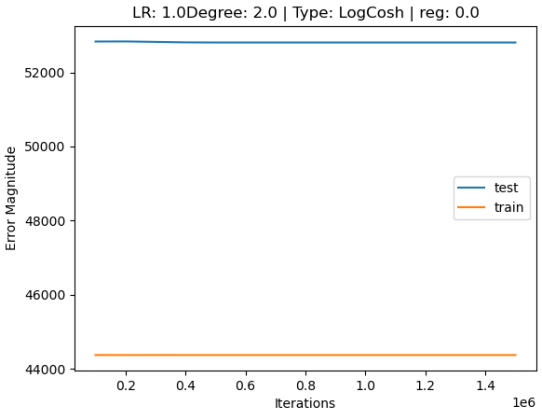


Figure 1.117: Degree 2

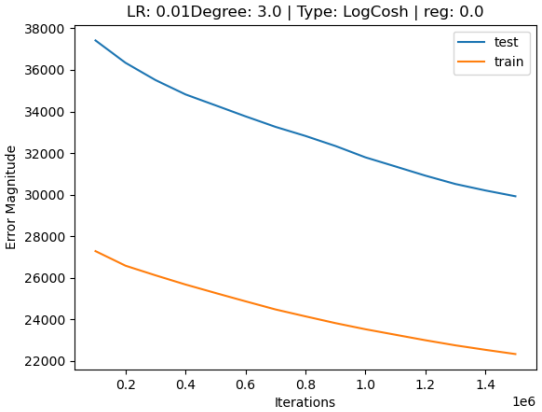


Figure 1.118: Degree 3

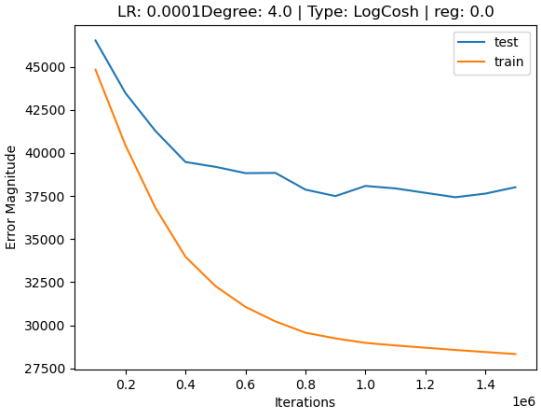


Figure 1.119: Degree 4

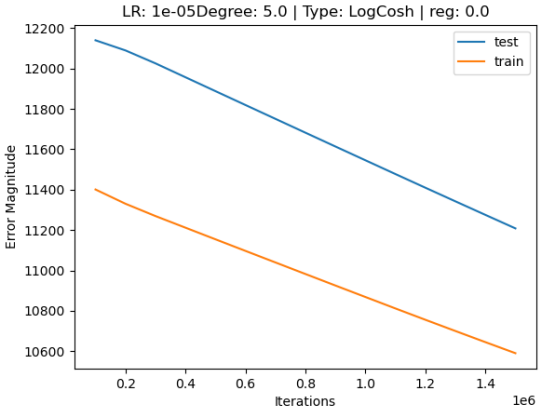


Figure 1.120: Degree 5

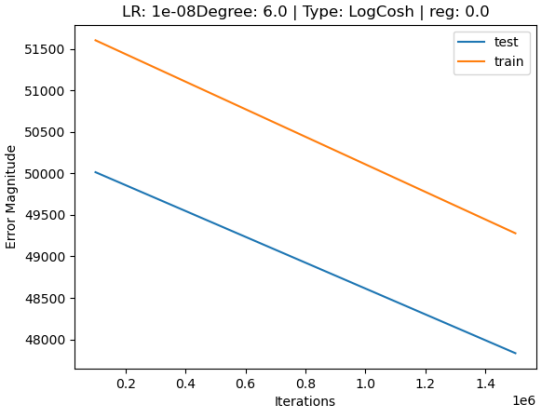


Figure 1.121: Degree 6

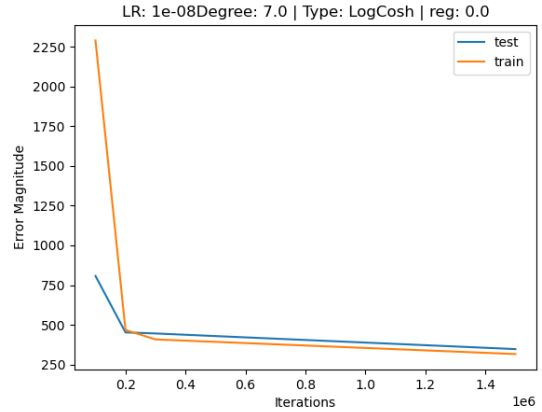


Figure 1.122: Degree 7

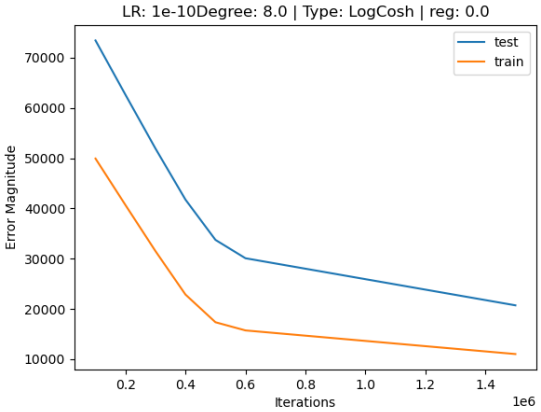


Figure 1.123: Degree 8

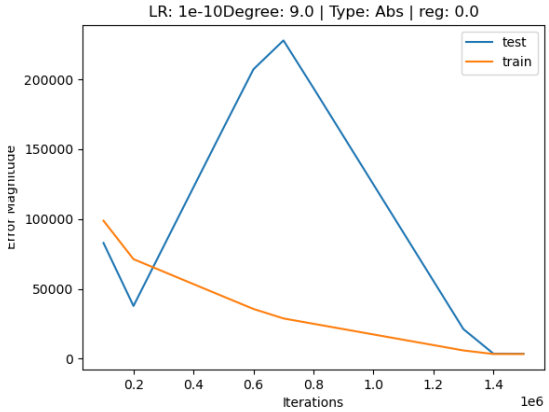


Figure 1.124: Degree 9

Interpretation: Graph 1

- There is a sudden rise in the training and testing errors for degree 6 which shows that degree 6 is a very poor fir on the data.
- Similarly, degree 8 also has a rise in the errors.
- The errors again decrease from degree 1 to 7(except at 6) after which they increase.
- The high training and testing error for degree 1 shows an under fit on the data.
- There is again a huge difference in the testing errors as compared to the the case of training on 20 data points supporting the observation that a 90:10 split is much better as compared to a 20:80 split.
- Both the error curves achieve a minima at 7 pointing out that 7 is the best degree.

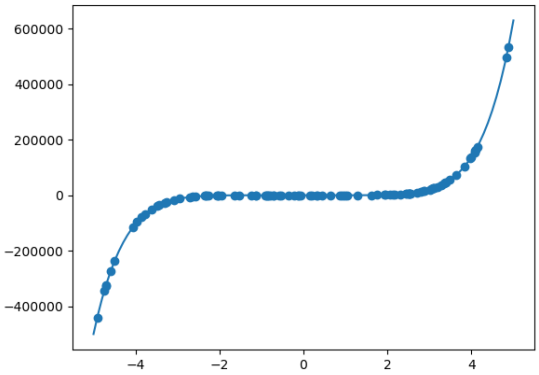
Interpretation: Graphs 2-10

- The graphs for degree 1 and 2 are very flat throughout because the model has already converged for a very small number of iterations.
- We observe a considerable difference in the training and testing error for degree 4,5 and 6 which could be because of an over fit on the training data.
- For degree 7 the model converges very quickly.
- For Degree 9 there is huge spike in the testing error which could be because of a high learning rate.

Conclusion:
Again degree 7 has the least errors and hence it is our guess for the best polynomial. So again we will use degree 7 for estimating the noise and guessing the underlying polynomial.

Estimation of noise variance and polynomial

To estimate the variance of noise we will use the fact that the mean is zero. $var(X) = E(X^2) - (E(X))^2$. Given E(X) = 0 we have $var(X) = E(X^2)$. Since X is the error hence this is equivalent to the mean square error value. So We have used that when the expected value of error is very close to 0 then the value of mean square error is the estimate for variance of noise.



Number of Iterations: 2000000
Training Error: 4.31
Testing Error: 4.56
Estimated Noise variance: 45.97
mean difference: 2.789
Degree: 7
Learning Rate: 10^{-7}

Figure 1.125: The estimated polynomial

Estimated Polynomial $3.99 + 2.15x + 5.33x^2 + 5.88x^3 + 3.39x^4 + 5.56x^5 + 4.00x^6 + 6.99x^7$

1.6 With Regularization - 90/10 - trained/tested

This section deals with the effects of regularization on our learning model on 90 random data points and testing on the remaining 10. Here, we have tried to find an optimum regularisation parameter for all degrees of polynomials from 1 to 9 and for all error functions as defined above.

1.6.1 Regularization parameter vs Error graphs for log-cosh function.

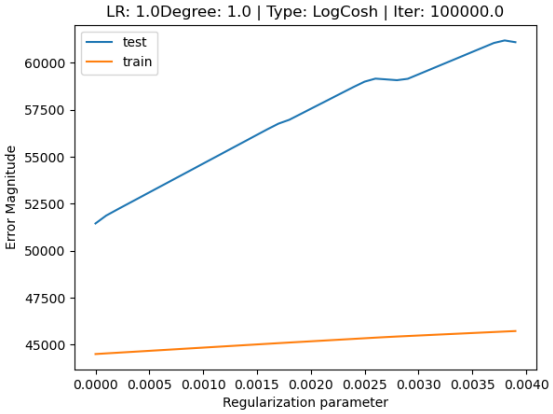


Figure 1.126: Degree 1

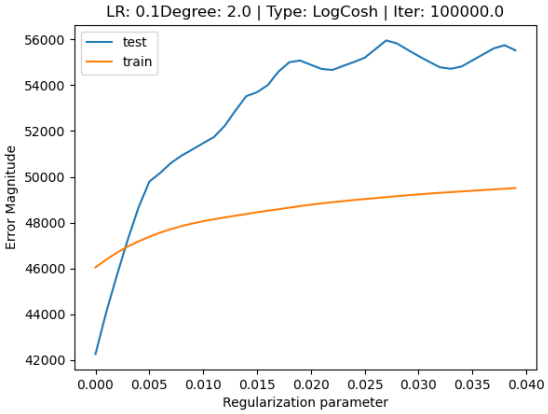


Figure 1.127: Degree 2

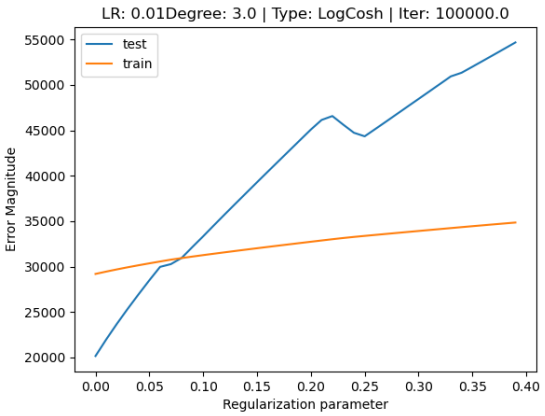


Figure 1.128: Degree 3

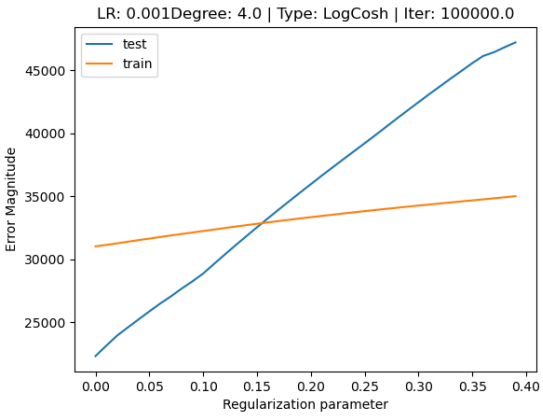


Figure 1.129: Degree 4

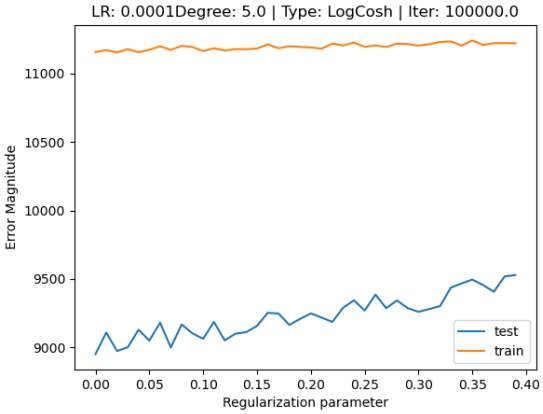


Figure 1.130: Degree 5

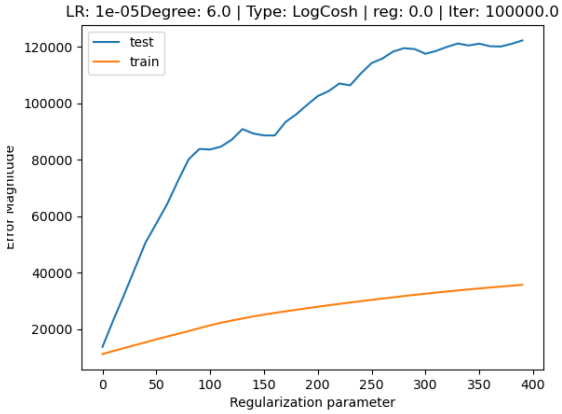


Figure 1.131: Degree 6

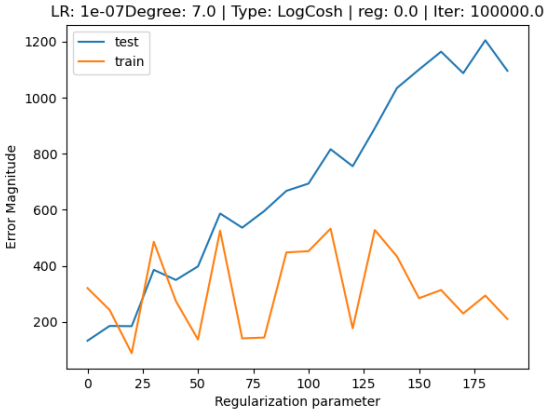


Figure 1.132: Degree 7

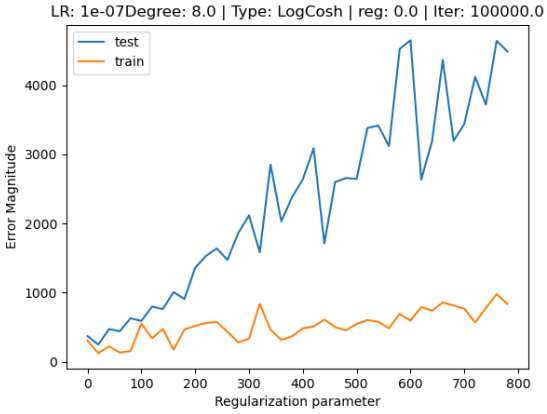


Figure 1.133: Degree 8

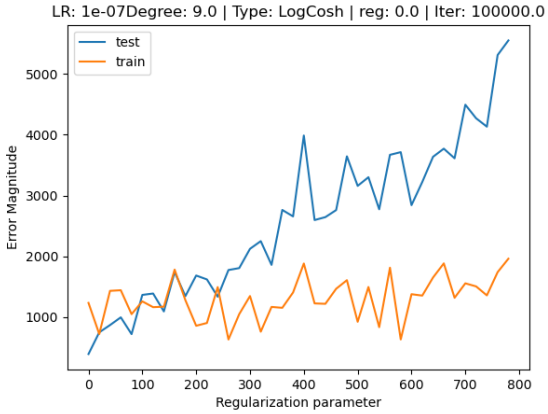


Figure 1.134: Degree 9

1.6.2 Regularization parameter vs Error graphs for mean absolute function.

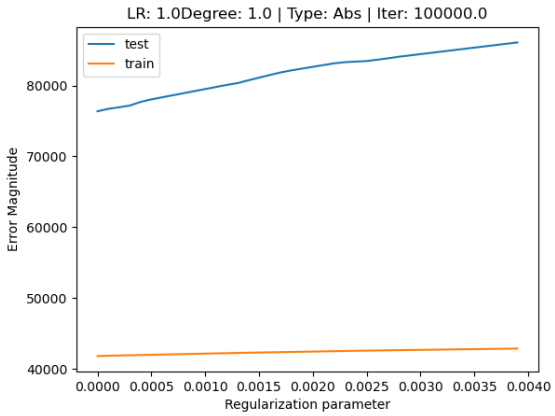


Figure 1.135: Degree 1

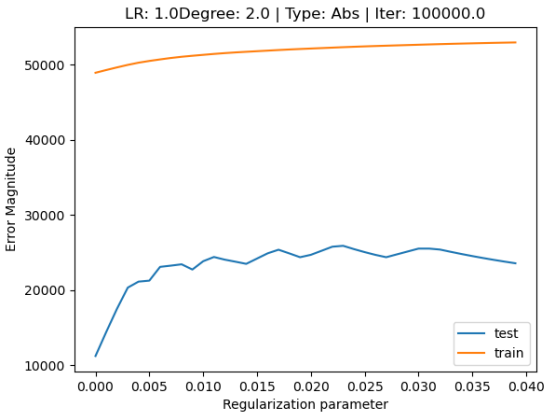


Figure 1.136: Degree 2

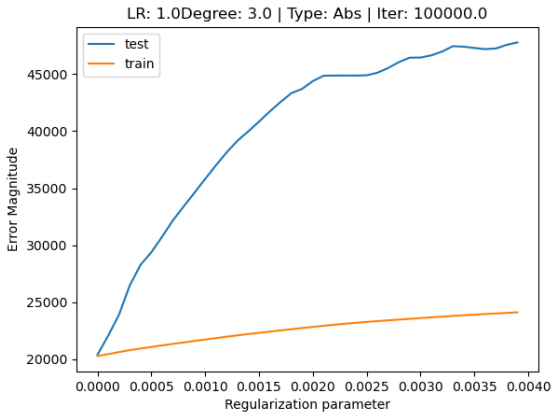


Figure 1.137: Degree 3

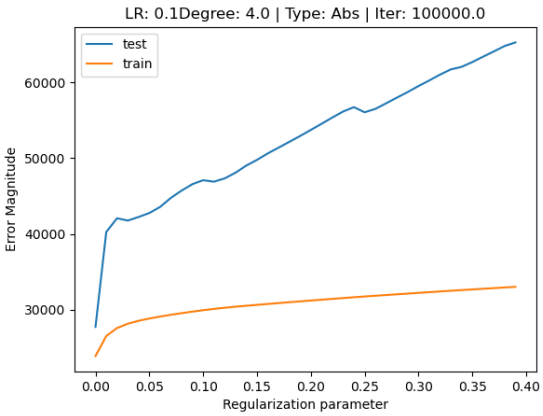


Figure 1.138: Degree 4

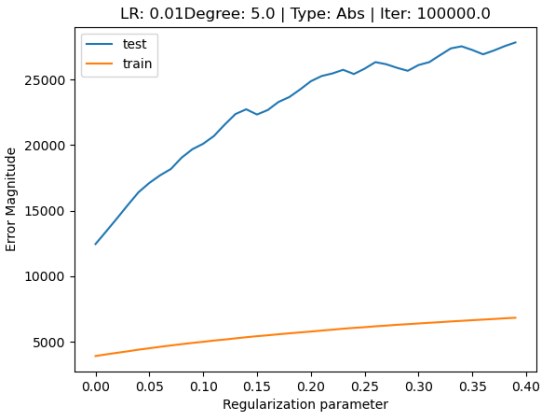


Figure 1.139: Degree 5

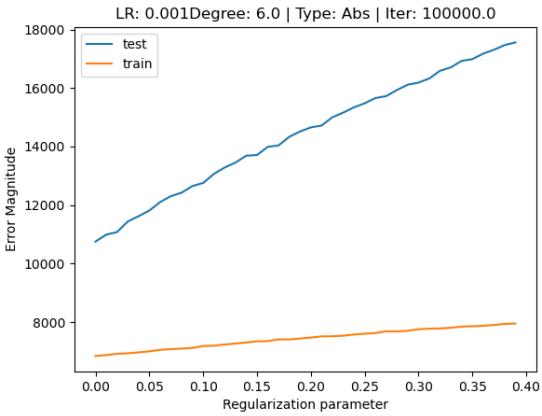


Figure 1.140: Degree 6

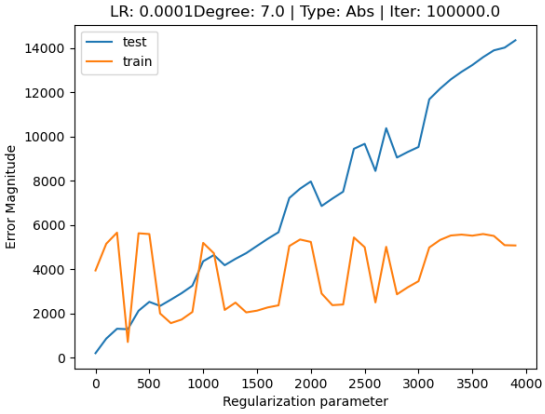


Figure 1.141: Degree 7

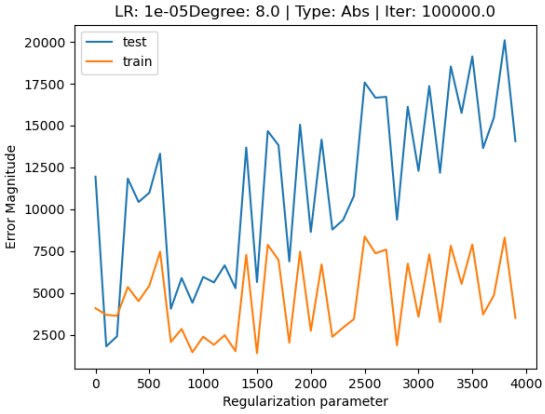


Figure 1.142: Degree 8

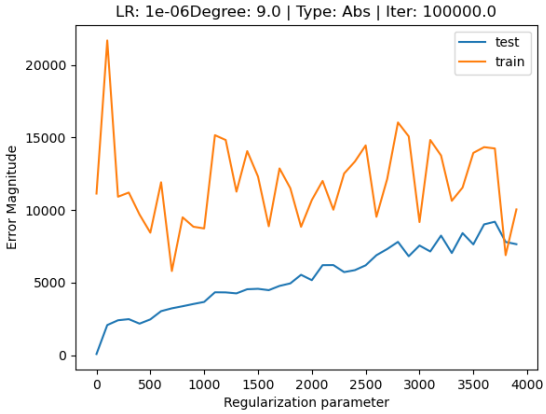


Figure 1.143: Degree 9

1.6.3 Regularization parameter vs Error graphs for log sigmoid error function.

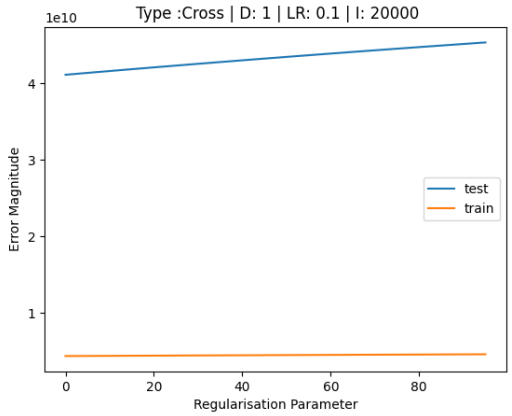


Figure 1.144: Degree 1

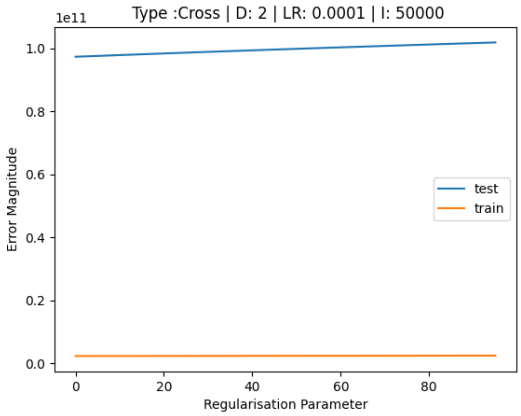


Figure 1.145: Degree 2

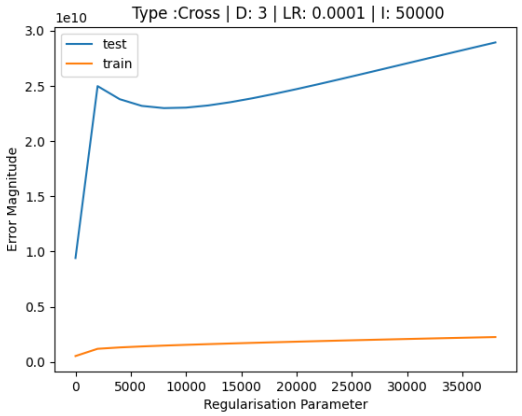


Figure 1.146: Degree 3

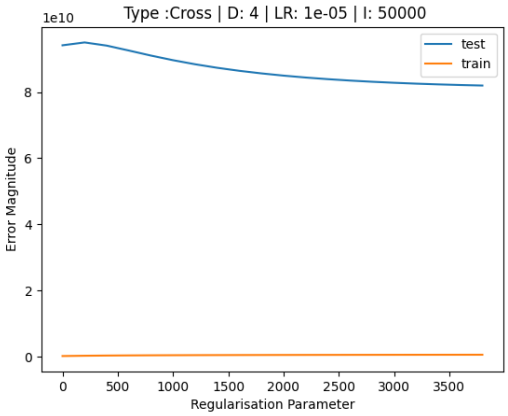


Figure 1.147: Degree 4

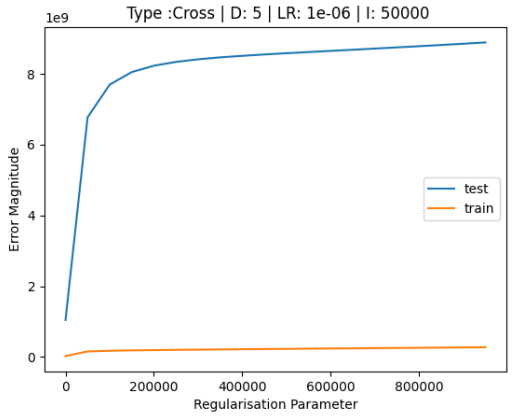


Figure 1.148: Degree 5

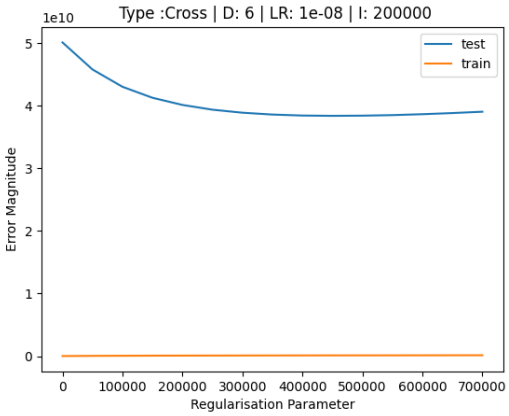


Figure 1.149: Degree 6

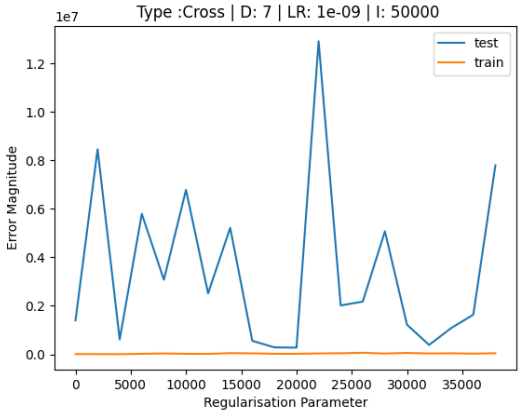


Figure 1.150: Degree 7

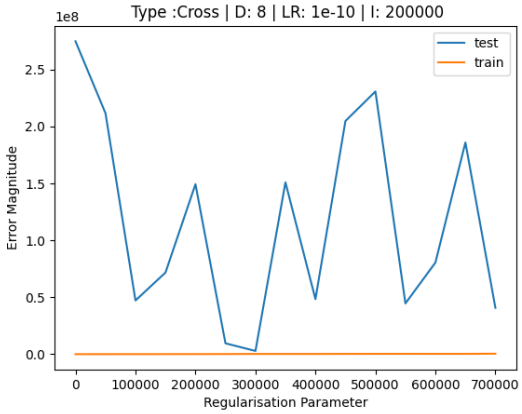


Figure 1.151: Degree 8

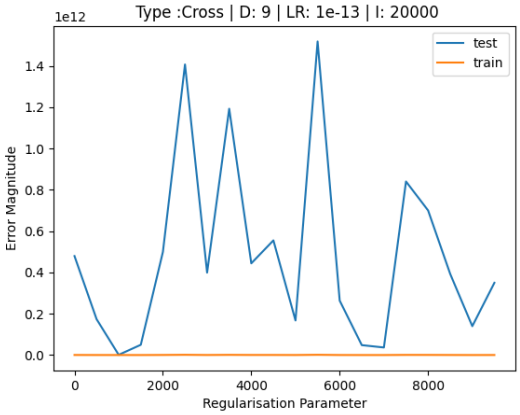


Figure 1.152: Degree 9

Conclusion: We observe that for the above 3 error functions introducing regularisation did not help much. As can be seen from the graphs above that more often than not the testing error increases on increasing the regularisation parameter. This can be explained by the graphs for these error functions without regularisation where most of the graphs do not show any strong evidence for over-fitting on the training data set. Moreover, this could also have been caused because of the very small size of the testing data set(10 data points) because of which a large error on even 1 outlier increases the average testing set error considerably.

1.6.4 Regularization vs Error graphs for Mean Root error function

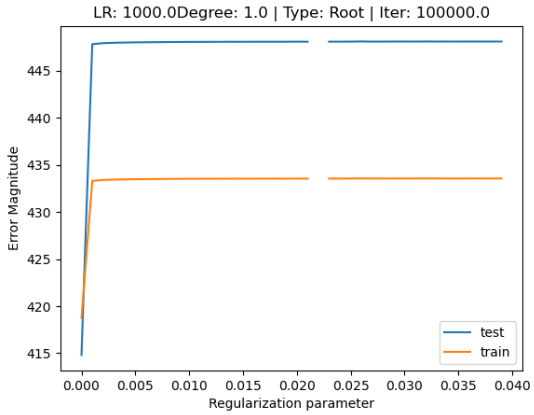


Figure 1.153: Degree 1

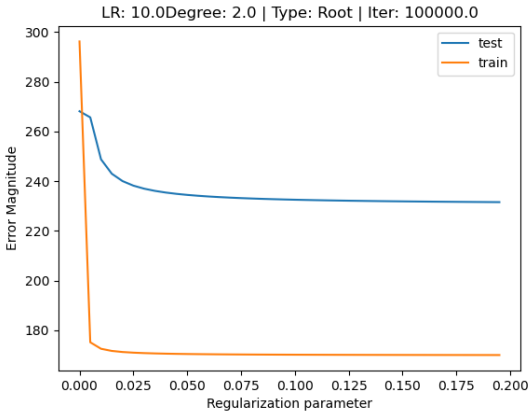


Figure 1.154: Degree 2

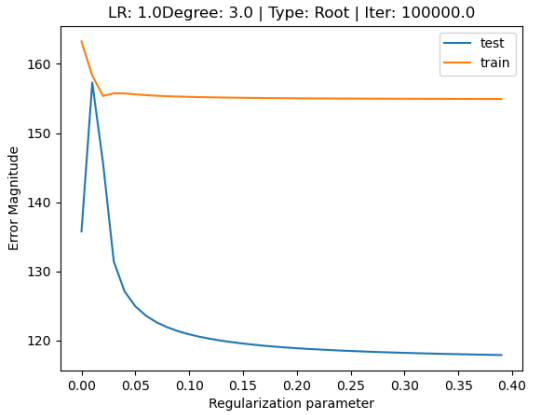


Figure 1.155: Degree 3

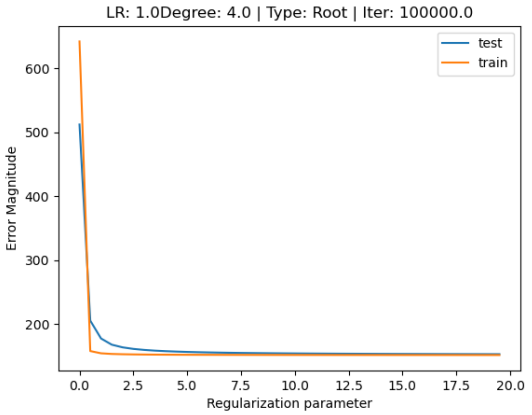


Figure 1.156: Degree 4

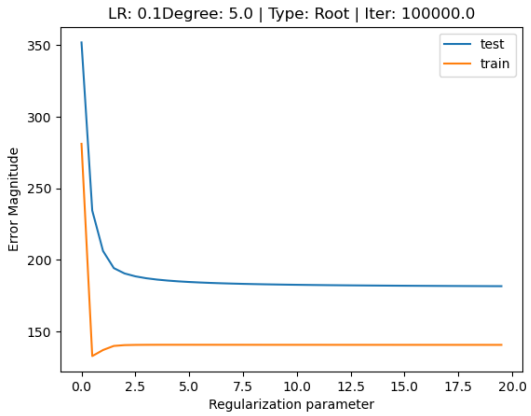


Figure 1.157: Degree 5

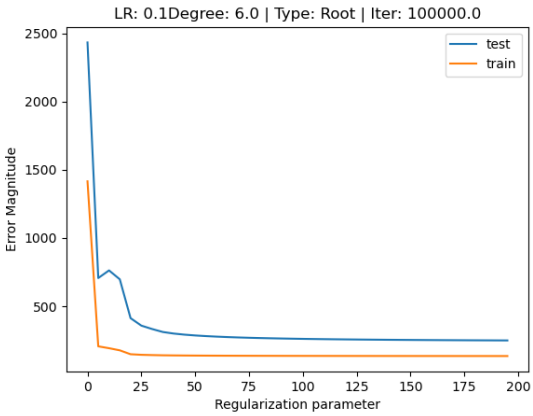


Figure 1.158: Degree 6

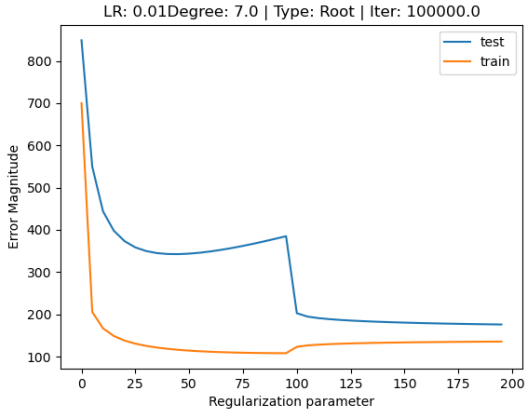


Figure 1.159: Degree 7

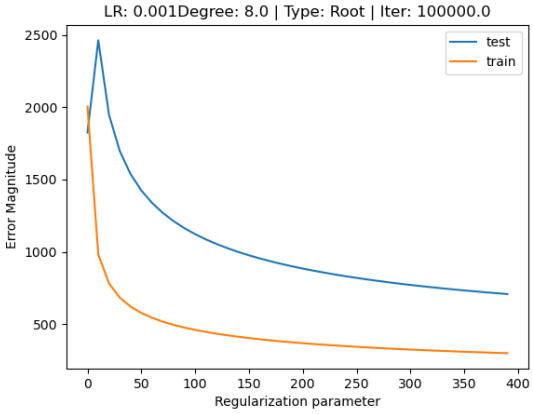


Figure 1.160: Degree 8

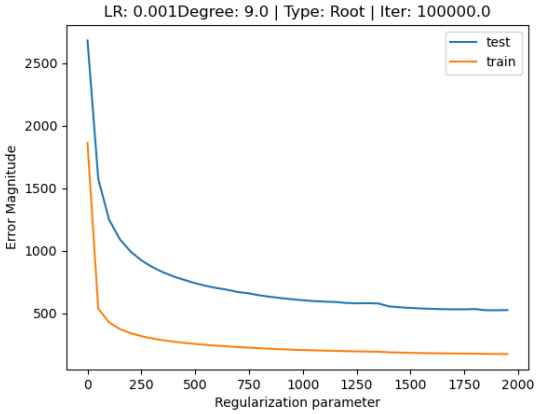


Figure 1.161: Degree 9

These graphs depict that there is a decrease in testing error and training error as we increase the regularization parameter for 100000 iterations, but we see the error is still very big compared to unregularized model. Hence we have created a table to portray that for which degree the testing error and training error decreased after regularization for 1500000 iterations. This table contains errors in case of regularized and unregularized model.

Degree	Regularization	Unreg Training	Unreg Test	Reg Testing	Reg Training
2	0.2	143	102	79	147
3	0.4	112	69.9	67	148
4	20	112	119	91	146
5	20	57.2	86.7	135	141
6	200	59.6	84.0	120	143
7	200	2.83	3.01	103	22
8	400	15.1	21.3	87	47
9	400	25.5	25.3	90	39

This is an interesting observation as we saw the test error to fall down as we increase the regularization parameter for 1,00,000 iterations, but when we do it for 15,00,000 iterations it falls down only in case of 2,3 and 4 degree while increases for others. This can be explained by the fact that regularization is causing a faster convergence. Since the above graphs are for 1,00,000 iterations, the unregularized model is yet to converge but the regularized model has already converged explaining decrease in test error in all cases. The following graphs reinforces the earlier convergence of regularized models for degree 3 and 4.

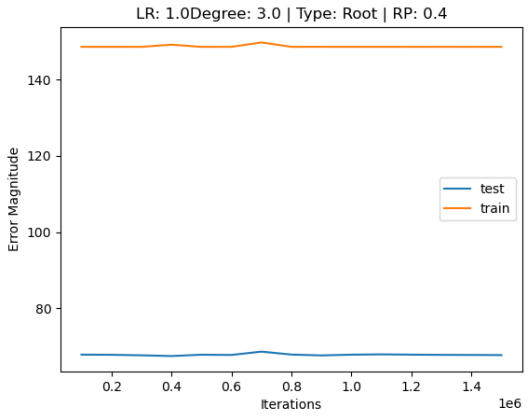


Figure 1.162: Degree 3 less iterations

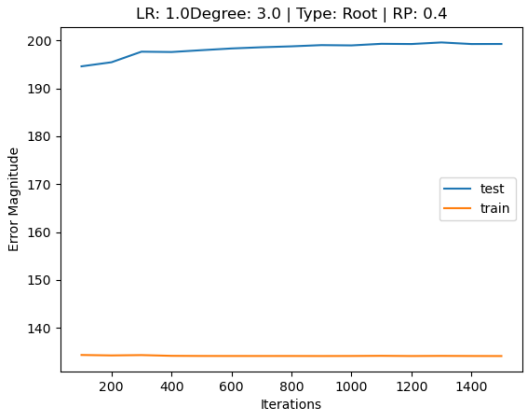


Figure 1.163: Degree 3

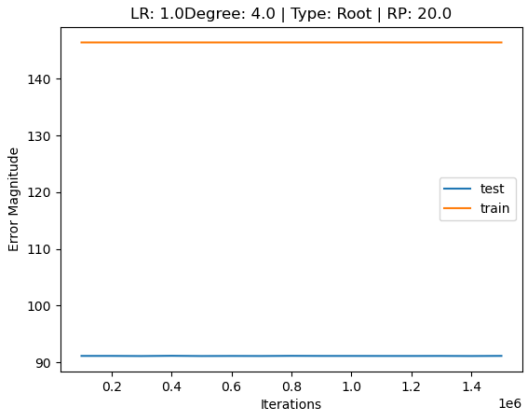


Figure 1.164: Degree 4 less iterations

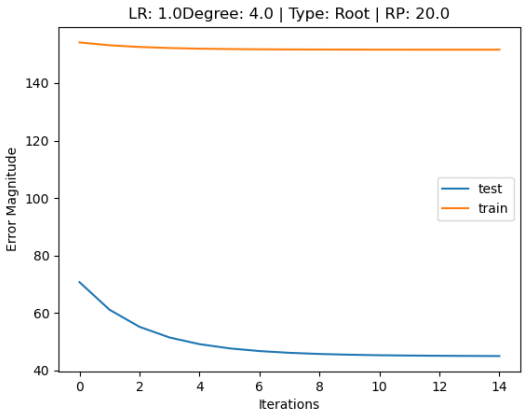


Figure 1.165: Degree 4

1.6.5 Conclusion:

We did not get any good enough regularization parameter that would indicate possibly a different degree of polynomial having the best fit. But we did see that it reduced over-fitting in many cases and also we noticed that it helped to converge faster to a value in case of root error function. This caused a confusion that it is better, but later we could see that it converged fast but at with lower accuracy. So in the end we conclude that no regularization parameter gives a conclusively better testing and training accuracy.

Chapter 2

Question 2

2.1 Training without regularization:

This section deals with training our model without regularization and observing accuracy of our model for various learning rates. This is used to determine the set of best learning rates, and also observe changes in accuracy with change in learning rate. The accuracy over iteration space is to determine the rate of convergence of model for a given learning rate.
We have used two different splits of the 5000 data points, 4000:1000 split (training:testing) and 3500:15000 split (training:testing).We have used the 3500:1500 split as well to allow the model to perform better on an unseen data set. We determine the best learning rates for each of these two splits, and then see their rate of convergence by graphing accuracy over iteration space.

2.1.1 4000:1000 split:

The testing accuracy is determined by the number of correctly guessed digits out of 1000 samples multiplied by 100. Similarly training accuracy is given by number of correctly guessed digits out of the 4000 samples multiplied by 100(i.e we have used classification error to check the accuracy). The following are the results for the 4000:1000 split when run for 2500 iterations.

Learning Rate	Training Accuracy	Testing Accuracy
0.001	25.38	25.0
0.1	88.3	85.3
1.0	95.8	88.0
1.5	96.88	86.6
5.0	96.52	86.8
10.0	97.92	86.4
50.0	96.03	85.3
100.0	91.97	82.4
1000.0	94.17	84.3
10000.0	94.92	85.2

We observe a very small accuracy when the learning rate is 10^{-3} as the learning rate because the model does not converge in 2500 iterations due to the small learning rate. Hence, we have added another graph without 10^{-3} to obtain a better look on the result for other learning rates.

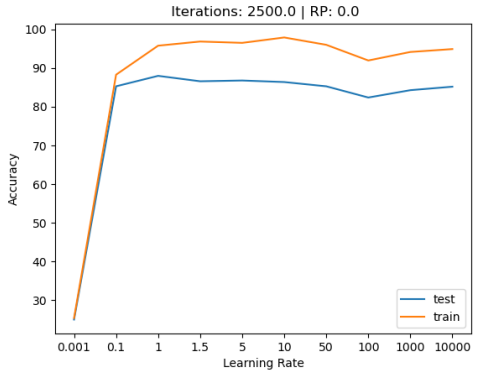


Figure 2.1: including 10^{-3} as learning rate

Figure 2.2: removing 10^{-3} as learning rate

Conclusion: From these graphs we conclude that the best learning rates are 1,1.5,5,10,50 and 10000 because as explained above 10^{-3} does not work well. Moreover, we observe a sharp dip in the accuracy for learning rate 100 and hence it is excluded for further analysis as well.

Accuracy vs Iterations for the above chosen learning rates:

Final training and test set accuracy after 4000 iterations:

Learning Rate	Training Accuracy	Testing Accuracy
1.0	96.9	87.5
1.5	97.7	87.4
5.0	96.38	85.3
10.0	96.13	85.0
50.0	94.6	85.0
10000.0	92.225	83.2

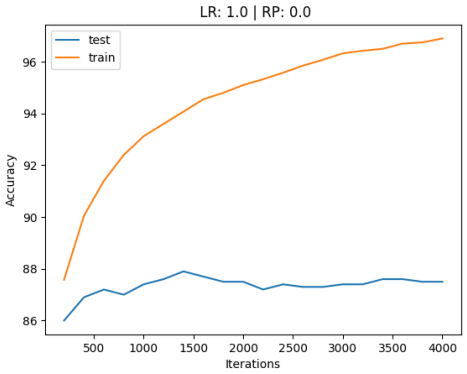


Figure 2.3: Learning rate : 1

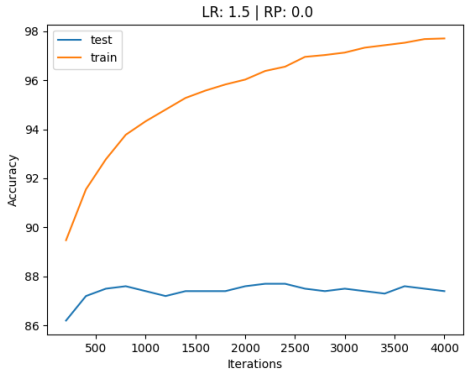


Figure 2.4: Learning rate : 1.5

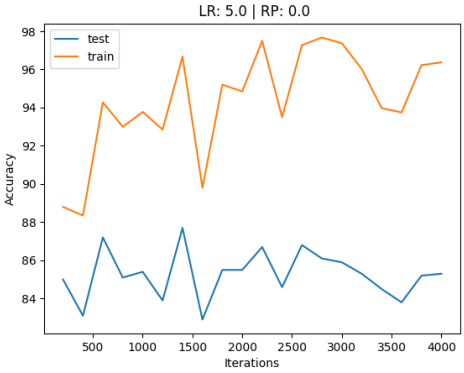


Figure 2.5: Learning rate : 5

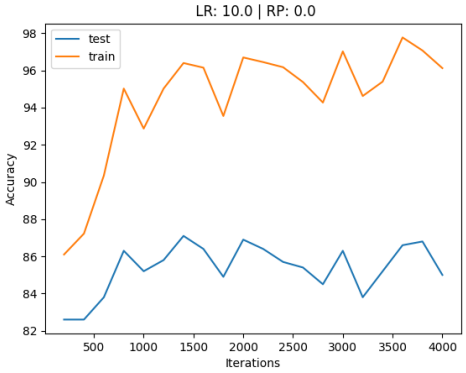


Figure 2.6: Learning rate : 10

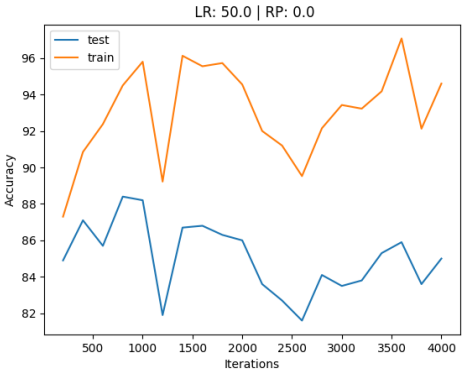


Figure 2.7: Learning rate : 50

Figure 2.8: Learning rate : 10000

Interpretations:

- We get smooth graphs for learning rates 1 and 1.5
- We get very inconclusive graphs for the other learning rates. This could be because the large learning rate does not allow convergence at the minima and we keep over shooting on either side of the minima with every iteration.
- The training and testing set accuracy starts decreasing sharply as we increase the learning rate above 5 because of the above mentioned reason.

Conclusions: The table above clearly suggests that among the chosen learning rates 1.5 is the best because it offers a very high testing set accuracy(87.4%) which is almost the highest we could get. We also get the highest training accuracy(97.7%) for learning rate 1.5

2.1.2 3500:1500 split:

The testing accuracy is determined by the number of correctly guessed digits out of 1500 samples multiplied by 100. Similarly training accuracy is given by number of correctly guessed digits out of the 3500 samples multiplied by 100(i.e we have used classification error to check the accuracy). The following are the results for the 3500:1500 split when run for 2500 iterations.

Learning Rate	Training Accuracy	Testing Accuracy
0.001	23.54	22.6
0.1	89.2	85.13
1.0	96.46	86.87
1.5	97.4	86.73
5.0	98.57	86.0
10.0	93.09	81.93
50.0	98.0	85.47
100.0	97.8	85.33
1000.0	97.74	85.8
10000.0	94.63	84.13

We again observe a very small accuracy when the learning rate is 10^{-3} as the learning rate because the model does not converge in 2500 iterations due to the small learning rate. Hence, we have added another graph without 10^{-3} to obtain a better look on the result for other learning rates.

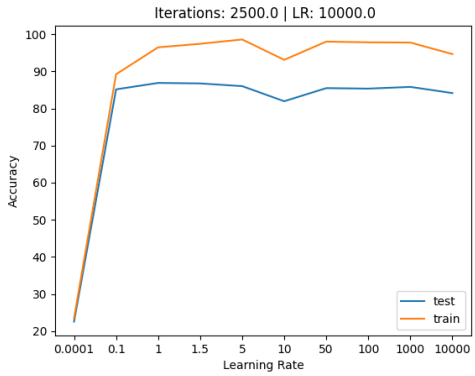


Figure 2.9: including 10^{-3} as regularization parameter

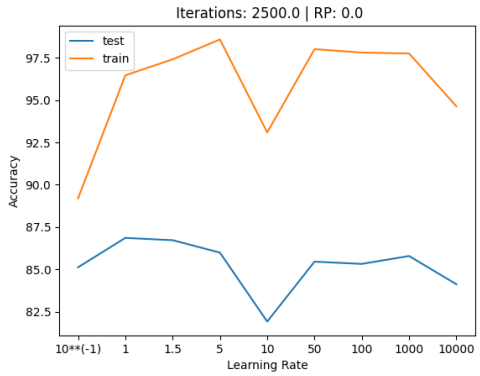


Figure 2.10: removing 10^{-3} as regularization parameter

Conclusion: From these graphs we conclude that the best learning rates for this split are 1,1.5,5,50,100 and 1000 because as explained above 10^{-3} does not work well. Moreover, the accuracy decreases sharply for learning rate 10 and hence it is excluded from further analysis.

Accuracy vs Iterations for the above chosen learning rates:

Final training and test set accuracy after 4000 iterations:

Learning Rate	Training Accuracy	Testing Accuracy
1.0	97.4	86.53
1.5	98.23	86.4
5.0	94.11	84.27
50.0	94.46	84.73
100.0	97.37	85.2
1000.0	97.71	85.2

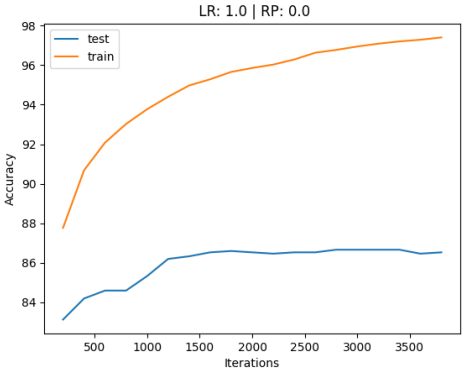


Figure 2.11: Learning rate : 1

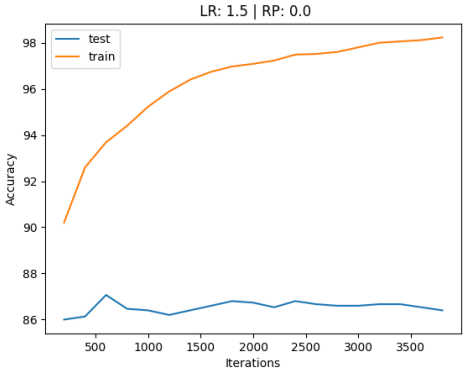


Figure 2.12: Learning rate : 1.5

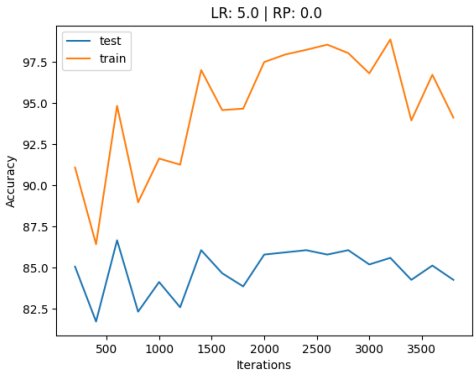


Figure 2.13: Learning rate : 5

Figure 2.14: Learning rate : 50

Figure 2.15: Learning rate : 100

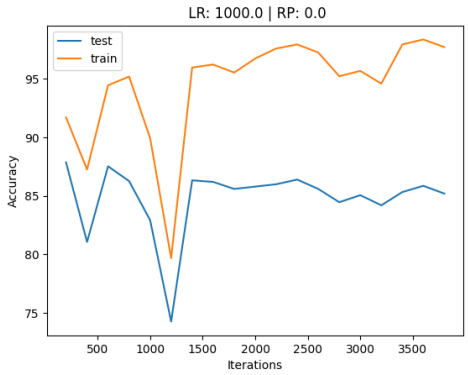


Figure 2.16: Learning rate : 1000

Interpretations:

- We get smooth graphs for learning rates 1 and 1.5
- We again get very inconclusive graphs for the other learning rates. This could be because the large learning rate does not allow convergence at the minima and we keep over shooting on either side of the minima with every iteration.
- The training and testing set accuracy starts decreasing as we increase the learning rate above 5 because of the above mentioned reason.
- When compared to the results of the 4000:1000 split, we observe a clear increase in the training set accuracy which could because of the smaller training data set size.

Conclusions: The table above clearly suggests that among the chosen learning rates 1.5 is the best because it offers a very high testing set accuracy(86.4%) which is almost the highest we could get. We also get the highest training accuracy(98.23%) for learning rate 1.5

2.2 Training with Regularization:

In this section we are going to introduce regularization in our model. We will compile test accuracy and training accuracy for the above chosen learning rates and over a range of regularization parameter to determine the best combination of the both. Once we determine the optimal combination for Regularization parameter and Learning rate, we will determine our best model for both 3500:1500 split and 4000:1000 split.

2.2.1 4000:1000 split:

We will use the learning rates found in the above section(i.e 1,1.5,5,10,50 and 10000) to calculate the testing and training accuracy. Moreover, we will use a range of regularization parameters(0.001,0.1,1,2,5,10,20,50 and 100) for each learning rate to find the optimum combination. Following are the results for each of the above mentioned learning rates.

Graphs:

Figure 2.17: Learning rate : 1

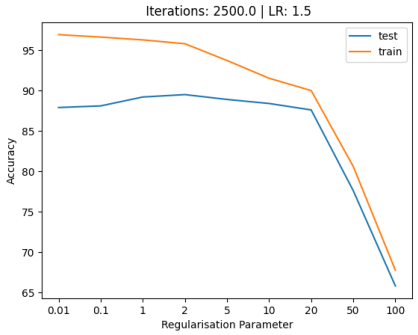


Figure 2.18: Learning rate : 1.5

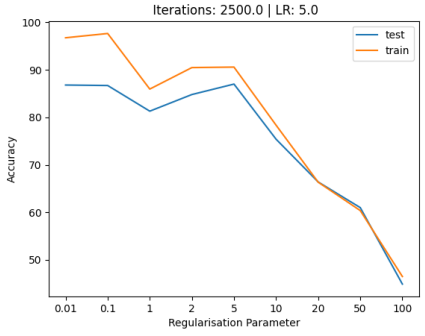


Figure 2.19: Learning rate : 5

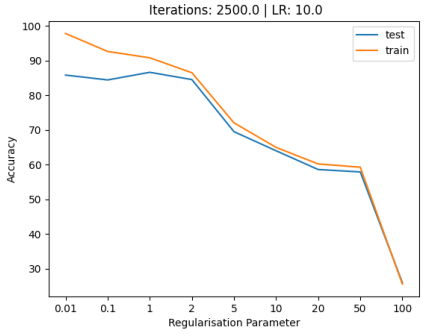


Figure 2.20: Learning rate : 10

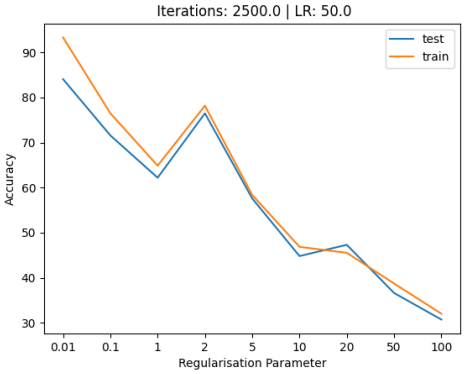


Figure 2.21: Learning rate : 50

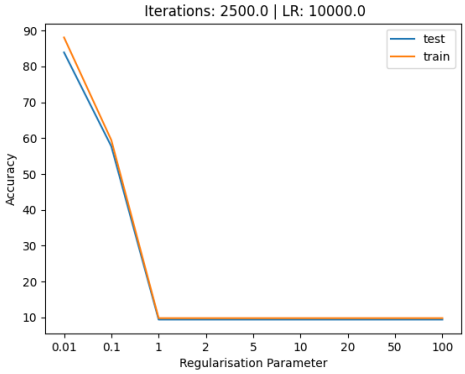


Figure 2.22: Learning rate : 10000

Heat maps:

Following is a heat map for testing and training accuracy over the range of learning rates and regularization parameters. This will help us have a better comparison across pairs of learning rate and regularisation parameters.

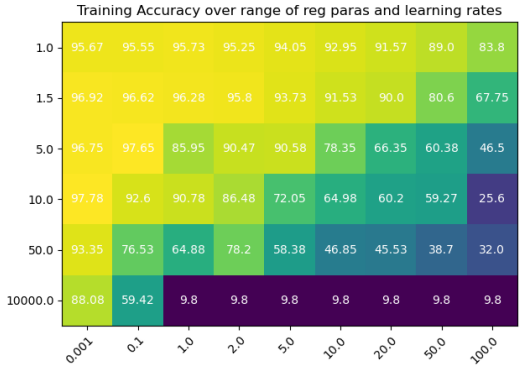


Figure 2.23: Training Accuracy

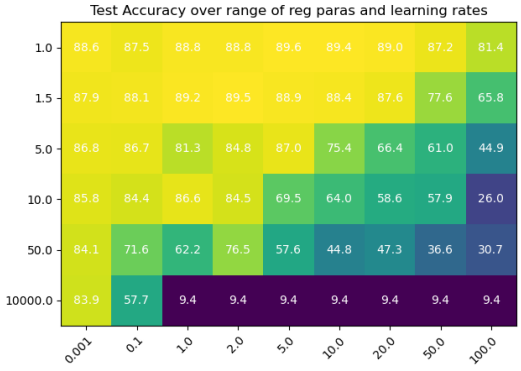


Figure 2.24: Testing Accuracy

Interpretations:

- For all learning rates, the accuracy more or less decreases on increasing the regularisation parameter.
- This is because increasing regularisation parameter implies strengthening the constraint on the weight vectors.
- This observation is also supported by the heat maps where the upper left and upper middle regions are the best.
- These regions correspond to a low regularisation parameter.
- In almost all graphs there is a sharp decrease in the accuracy after regularisation parameter crosses 50.
- It would be irrational to comment anything on the shape of the graph because they are not drawn to scale, rather a discrete set of points are simply plotted.

Next, for each learning rate we have chosen the best regularisation parameter and run the model for larger number of iterations. Following are the results on running the model for 4000 iterations.

Learning Rate	Regularisation Parameter	Training Accuracy	Testing Accuracy
1.0	0.01	96.92	88.1
1.5	2.0	95.97	89.2
5.0	0.1	92.0	83.1
10.0	0.01	95.67	84.6
50.0	0.01	94.0	83.9
10000.0	0.01	86.67	83.5

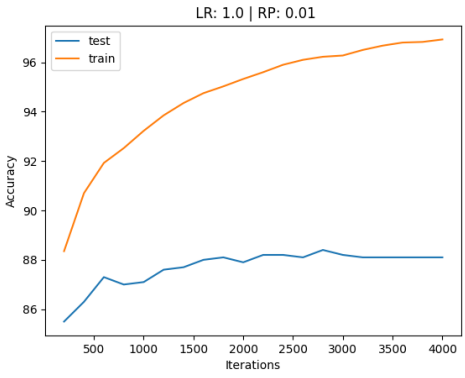


Figure 2.25: Learning rate : 1

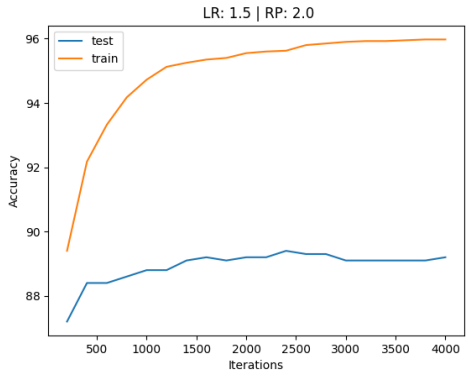


Figure 2.26: Learning rate : 1.5

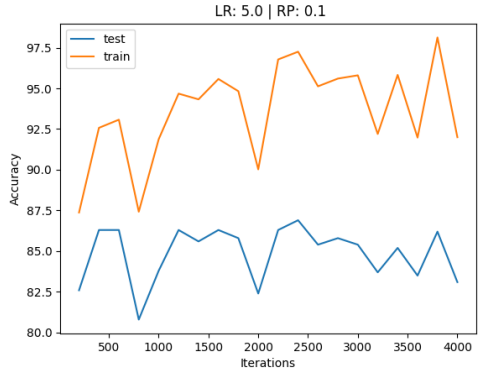


Figure 2.27: Learning rate : 5

Figure 2.28: Learning rate : 10

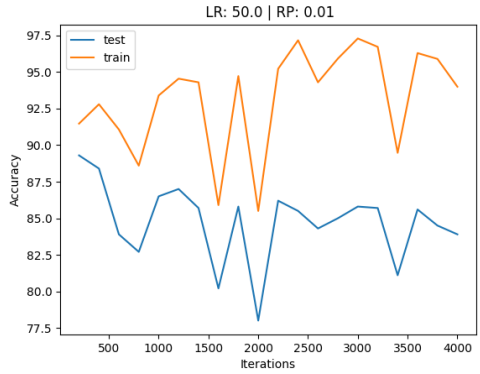


Figure 2.29: Learning rate : 50

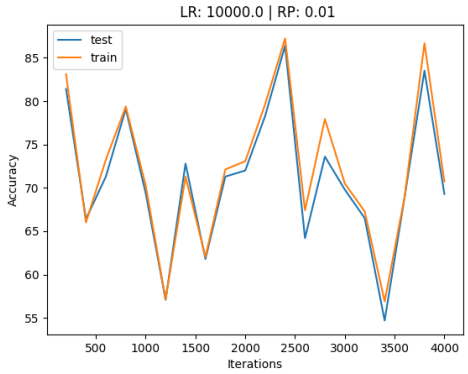


Figure 2.30: Learning rate : 10000

Conclusions:

We again get smoother graphs for smaller learning rates.
On comparing these results with the one without regularisation, we observe an increase in the training set accuracy which is a result of decrease in the over fit on the training set.
Based on the observations above the model we choose is the one with learning rate 1.5 and regularisation parameter 2.0 because we get the highest testing set accuracy(89.2%) using it without much decrease in the training accuracy(95.97%)

2.2.2 3500:1500 split:

We will use the learning rates found in the above section(i.e 1,1.5,5,50,100 and 1000) to calculate the testing and training accuracy. Moreover, we will use a range of regularization parameters(0.001,0.1,1,2,5,10,20,50 and 100) for each learning rate to find the optimum combination. Following are the results for each of the above mentioned learning rates.

Graphs:

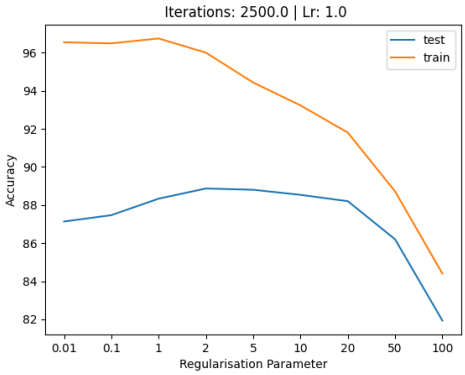


Figure 2.31: Learning rate : 1

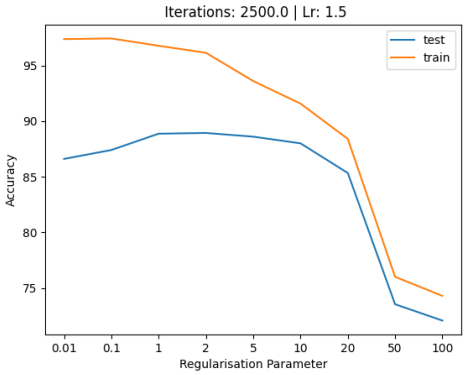


Figure 2.32: Learning rate : 1.5

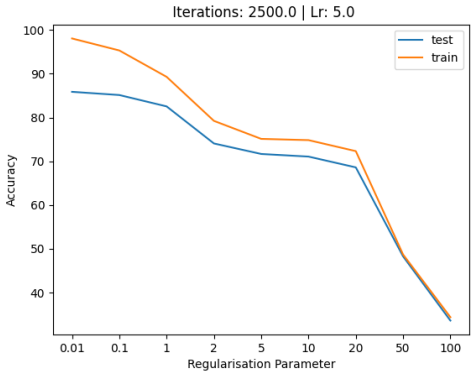


Figure 2.33: Learning rate : 5

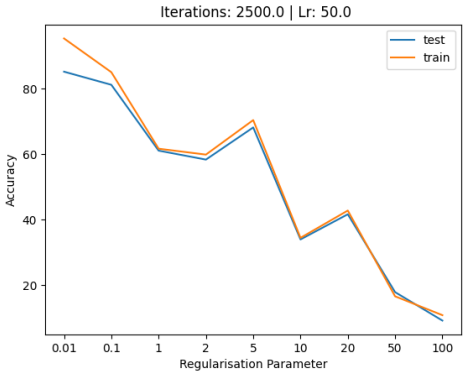


Figure 2.34: Learning rate : 50

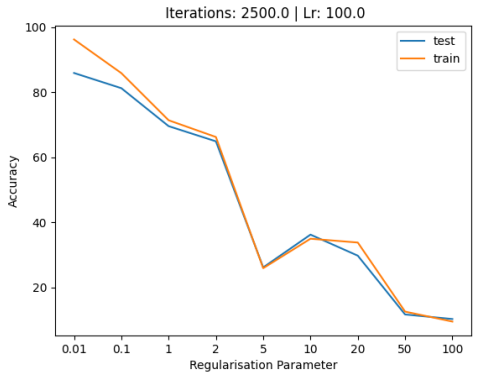


Figure 2.35: Learning rate : 100

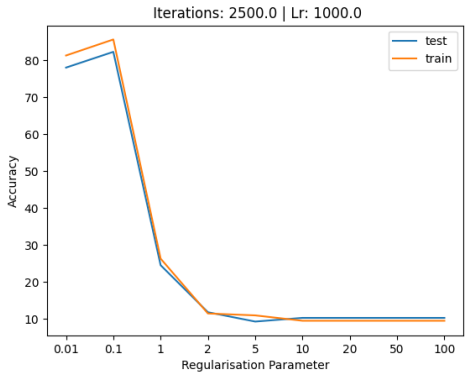


Figure 2.36: Learning rate : 1000

Heat maps:

Following is a heat map for testing and training accuracy over the range of learning rates and regularization parameters. This will help us have a better comparison across pairs of learning rate and regularisation parameters.

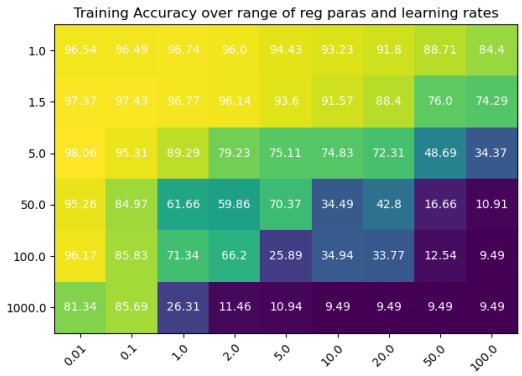


Figure 2.37: Training Accuracy

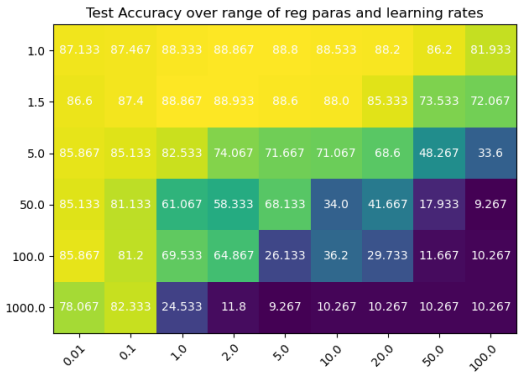


Figure 2.38: Testing Accuracy

Interpretations:

- For all learning rates, the accuracy more or less decreases on increasing the regularisation parameter.
- This is because increasing regularisation parameter implies strengthening the constraint on the weight vectors.
- This observation is also supported by the heat maps where the upper left and upper middle regions are the best.
- These regions correspond to a low regularisation parameter.

- In almost all graphs there is a sharp decrease in the accuracy after regularisation parameter crosses 50.
- It would be irrational to comment anything on the shape of the graph because they are not drawn to scale, rather a discrete set of points are simply plotted.

Next, for each learning rate we have chosen the best regularisation parameter and run the model for larger number of iterations. Following are the results on running the model for 4000 iterations.

Learning Rate	Regularisation Parameter	Training Accuracy	Testing Accuracy
1.0	2.0	96.17	88.93
1.5	1.0	97.14	87.8
5.0	0.01	99.03	85.87
50.0	0.01	90.66	81.73
100.0	0.01	94.6	83.67
1000.0	0.1	73.74	71.0

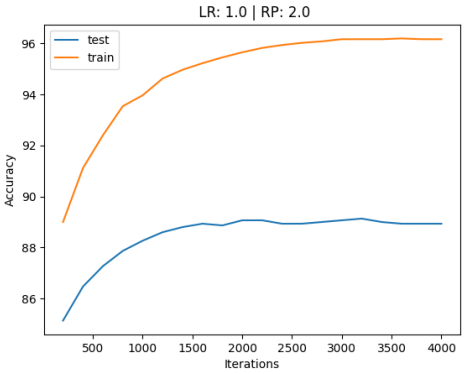


Figure 2.39: Learning rate : 1

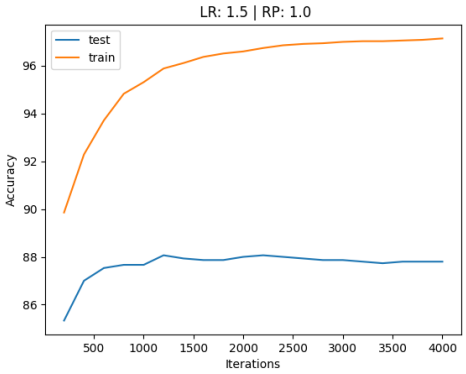


Figure 2.40: Learning rate : 1.5

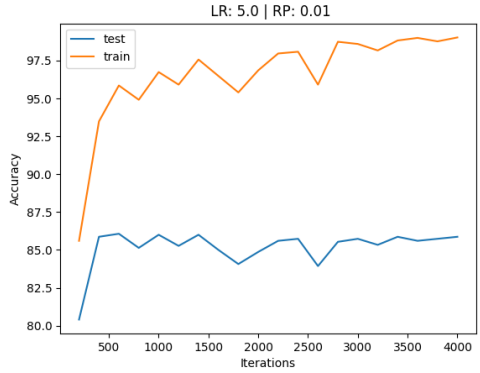


Figure 2.41: Learning rate : 5

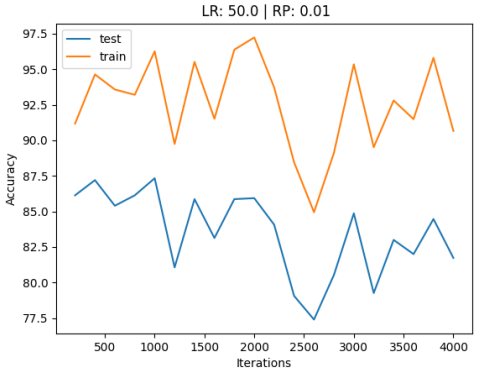


Figure 2.42: Learning rate : 10

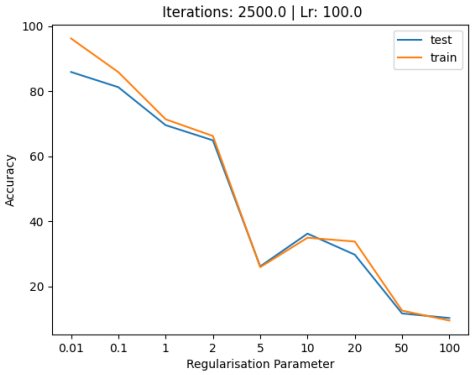


Figure 2.43: Learning rate : 50

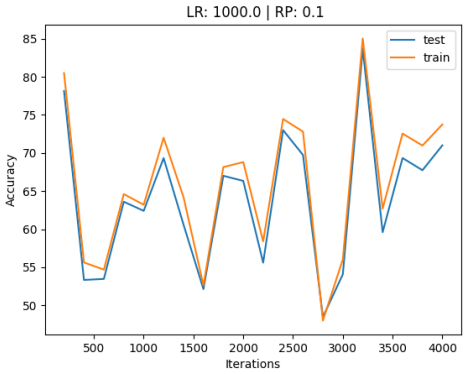


Figure 2.44: Learning rate : 10000

Conclusions:

We again get smoother graphs for smaller learning rates.
On comparing these results with the one without regularisation, we observe an increase in the training set accuracy which is a result of decrease in the over fit on the training set.
On comparing these results with the results obtained for the 4000:1000 we observe an increase in the training and a decrease in the testing set accuracy.
Based on the observations above the model we choose is the one with learning rate 1.0 and regularisation parameter 2.0 because we get the highest testing set accuracy(88.93%) using it with a high training accuracy(95.97%) as well.