REPORT ON ASSIGNMENT-4

CS669-Pattern Recognition

Submitted By: Group 18

Submitted to: Dr. A.D Dileep

Members:

Abhishek Singh Rawat (T23191)

Arpit Dua (T23192)

Rohit Roy (S22048)

Introduction:

- The task at hand is regression. In statistical modelling, regression analysis is a set of statistical process for estimating the relationship between a dependent and independent variable (often called the 'outcome' or 'response' variable , or a 'label' in machine learning parlance) and one or more independent variables (often called 'predictors'/'features'.
- Regression analysis is primarily used for prediction and forecasting, where it is closely related to the field of machine learning.
- The kind of data which we work on has labels as continuous values rather than discrete labels as in classification.
- Regression is a supervised learning technique.
- From a high view we have to fit a function on data which will be an approximation of the real function which results in that data and that approximation should result in the minimum error between the predicted model output and the actual output of the system.

System overview

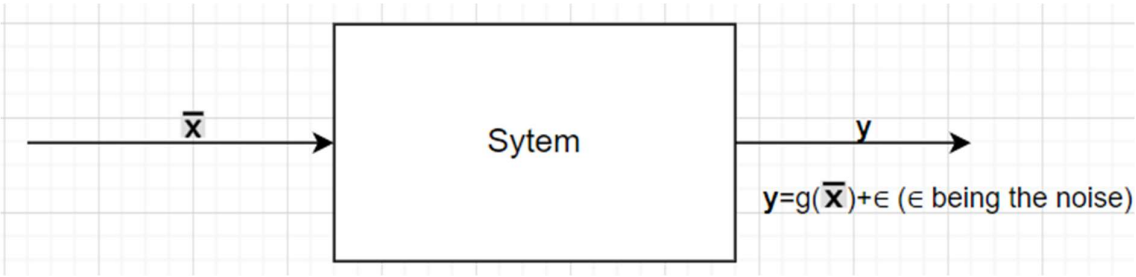


Fig1

Real physical systems might have noise in the data.

For a single input variable like years of experience (an independent variable acting as one of the features) for predicting salary (a dependent variable) we might obtain different salaries as output which might be due to the non-consideration of the other independent features since the output should be a cumulative effect of the independent variables.

Model overview

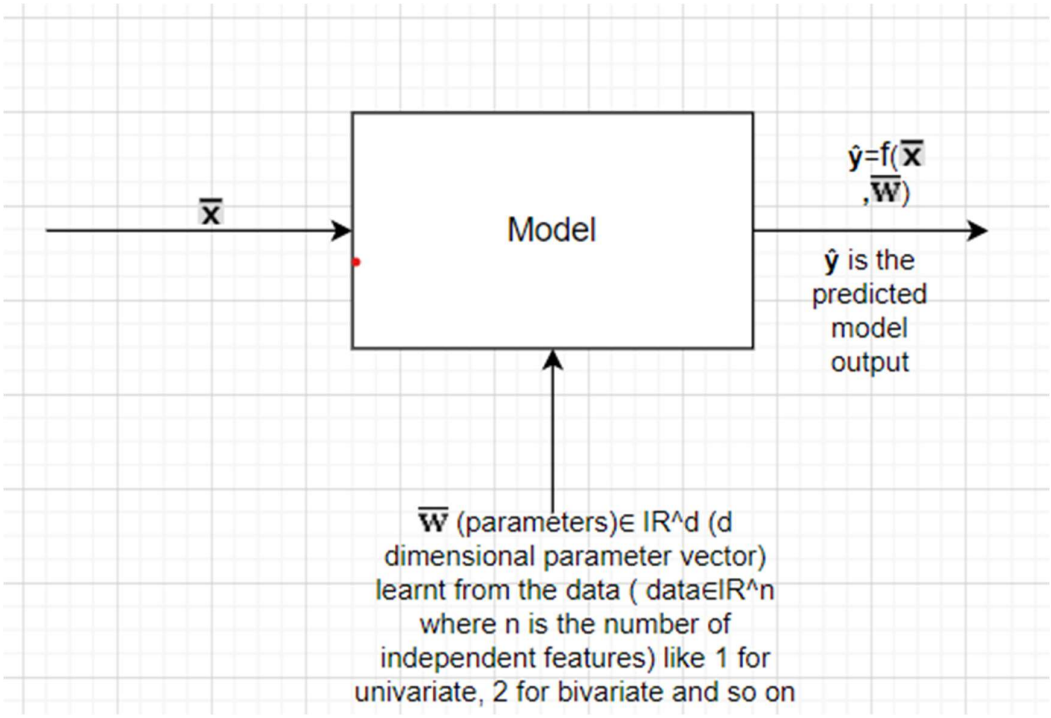


Fig2

The predicted model output should be as close as possible to the system output.

For that to happen we need to make some assumptions about the nature of the function f (varying from case to case).

We have to learn the parameters of the model from the data such that error function is minimised w.r.t. $\overline{w}$ parameters.

$E(\overline{w}) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$

The above equation can be written as:

$E(\overline{w}) = \frac{1}{2} \sum_{n=1}^{N} (y_n - f(x_n, \overline{w}))^2$

Depending on certain factors like: -

1) Number of training samples available.
2) Dimension of the input data (like univariate, bivariate etc).

We can choose the kind of polynomial and its complexity (higher degree referring to higher complexity) as the function for our model.

Results/Observations:

Polynomial curve fitting for Dataset 1

Dataset1

Characteristics of the dataset 1:

- Data in a csv file (.csv format).
- Data is univariate which means only one independent feature (x).
- We have the corresponding outputs to the inputs.
- Dataset has 1001 entries.
- The values of x lie between 0 and 1 (they being included).
- The y values lie between -0.14576 and 4.1118.

After a 70-30 split we get the train data as 700 entries and test as 301 samples.

Procedure: -

1. Starting our curve fitting for 700 samples and moving onto a lower number of samples (100, 50 and finally 10 training samples).
2. We know from theory that when we want to fit a **m** degree curve for our univariate data we need around **10*m samples** as a thumb rule for training/learning and for our model function to be a good approximation of the actual system function.
3. This happens because the more complex the model (model is another way of saying that polynomial/function which we want to fit) the more is the number of polynomial coefficients/parameters to be estimated and hence we need more data to learn them.
4. So we start with curve fitting for 700 train samples which will include fitting a curve from degree 2 (equation being w0+w1*x+w2*x^3 i.e. 3 parameters) to degree 9 (equation being w0+w1*x+w2*x^2 +w3*x^3+w4*x^4+w5*x^5+w6*x^6+w7*x^7+w8*x^8+w9*x^9 i.e. 10 parameters).

   Polynomial curve fitting algorithm explanation starts from the next page.

5. Fitting a polynomial curve for a certain number of samples has a standard algorithm which is as follows:

   When our data is univariate and we want to fit a d degree curve to a training data having n samples we calculate our **design/vandermonde matrix.**

$$X = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^d \\ 1 & x_1 & x_1^2 & \cdots & x_1^d \\ & \vdots & & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^d \end{bmatrix}_{n*(d+1)}$$

   Fig3

   Now we obtain a pseudo inverse matrix **X**+ =((**X**^T * **X**)^-1)* (**X**^T)

   * $\bar{y}$

   Where [**X**^T] (d+1) * n is the transpose of the design matrix.

   On multiplying it with **X** we get a matrix of the dimension (d+1)*(d+1) and taking its inverse (assuming it exists) makes it again a (d+1)*(d+1).

   Now we multiply (**X**^T *X) ^-1 which is a (d+1) *(d+1) matrix with we multiply **X**^T which is a (d+1) * n matrix which is essentially the dimension of the design matrix.

   X+ is a matrix **[X+]** (d+1) *n

   Our $\bar{y}$ is a n*1 vector consisting of all the outputs stored as a single column vector corresponding to each x's in our training examples.

   Our final answer which is the parameter vector should be of the size

   We know that since our degree of the polynomial was assumed to be d in the beginning our polynomial should look like w0+w1*x+w2*x^2+.....+wd-1*x^d-1+ wd*x^d i.e if we make a vector for the parameter values we will get something like

   Predicted Parameter vector ($\hat{w}$)= [w0 w1 w2 w3…wd-1 wd] ^T (^T denoting transpose). Hence the dimension of $\hat{w}$ is (d+1) *1.

   If we multiply our pseudo inverse matrix with $\bar{y}$ we get our final parameter vector.

   [$\hat{w}$] (d+1) *1 = **[X+]** (d+1) *n * [$\bar{y}$]n*1

Now using the parameters we can interpolate for each point within our training data and find the corresponding predicted train output.

After the step above we are in a position to calculate train MSE (mean squared error).

Now we use our parameters to get the predicted test outputs at all the test sample points in the input.

After the above step we are in a position to calculate test MSE (mean squared error).

We plot the train and test MSE on the same graph having varying complexities/degrees of polynomial on the x axis and train and test MSE values on y axis.

After observing the phase where there is a significant gap in the values of the train and test MSE which essentially means observe those degrees/complexity of the model where our model is overfitted (train loss is low and validation/test loss is high) which could be because our model tried learning noise along with the pattern. Since the noise is stochastic in nature it results in low training loss but high validation loss.

After that we need to perform regularisation for that particular degree and above where our model is overfitted.

This means we try to minimise both the training and testing loss simultaneously, restricting our model's parameters/coefficients to smaller values indicating mitigation of overfitting problems. It is kind of like finding a sweet spot between overfitting and loss.

Re estimated parameter vector after regularisation where λ is a hyperparameter (regularisation parameter) is as mentioned below

$\hat{w}$ =((**X**^T * **X**- λ I)^-1)* (**X**^T)* $\bar{y}$

After regularisation for each degree we plot the graph for values of ln λ on the x axis which will vary from -infinity to 0 (as λ varies from (0, 1]) and train and test loss on the y axis corresponding to each value of lambda. This happens because

Varying the value of λ - - - - - > we get different $\hat{w}$ /parameter values - - - - -> this results in different predicted train and test graphs and hence the corresponding values for train and test MSE and their graphs.

Finally for a particular degree we select an ideal value of λ and calculate the final predicted train and test outputs, corresponding MSE and their respective graphs.

Some miscellaneous points: -

1. At all the places the model that we are building will be termed as **linear model of regression** because our model is always linear in terms of the coefficients/parameter values.

2. When we perform r**egularisation framed as a constraint optimization problem in terms of L2 norm** as we will do in the case of univariate and bivariate data we call it **ridge regression**.

Explanation:

What is the Ridge Regression?

Also known as Ridge Regression L2. In another source, it is defined as follows. The main purpose of ridge regression is to find the coefficients that minimise the sum of error squares by applying a penalty to these coefficients.

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularisation. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

Ridge regression is the regularised form of linear regression.

Features of Ridge Regression

In Ridge Regression, the model is set up with all variables given. However, it does not remove variables with low relationships from the model, it brings the coefficients of these variables closer to zero.

- It is resistant to overlearning.
- It is biassed but has a low variance.
- It is better than the Least Squares method when there are too many parameters.
- It offers a solution against multidimensionality. The problem here is that the number of variables is greater than the number of observations. It offers a solution against this.
- It is effective in multiple linear connection problem. The problem here is that there is a high correlation between the independent variables.
- It is important to find an optimum value for λ. Cross-Validation is used for this.

Lambda → λ. In Ridge Regression, λ plays a critical role. It allows controlling the relative effects of the two terms. So, actually λ is the penalty term. Given λ is represented as an alpha parameter in the Ridge Regression function. By changing the alpha value, we control the penalty term. If λ is zero, this gives us the classical regression equation. Consequently, the higher the Alpha values, the greater the penalty. Therefore, the size of the coefficients is reduced.

- It shrinks the parameters. Therefore, it is used to prevent multicollinearity

- It reduces the model complexity by coefficient shrinkage

Ridge Regression Model

Ridge Regression Model is a version of the classical regression equation with a correction function.

$$SSE_{L_2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{P} {\beta_j}^2$$

Ridge Regression SSE Formula

The left side of the equation expresses the classical regression calculation. On the right, each Beta value is squared. And these values add up. Then the model is standardised by multiplying λ by the setting parameter. We can call this correction.

Steps of Ridge Regression

- The λ setting parameter is determined by the user.
- Beta coefficients are calculated from the data set.
- A set containing specific values for λ is selected. And Cross-Validation test error is calculated for each.
- The λ, which gives the smallest Cross-Validation, is chosen as the setting parameter.
- Finally, the model is re-fitted with this λ selected.

- **Overfitting**

  In mathematical modelling, overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit to additional data or predict future observations reliably". An overfitted model is a mathematical model that contains more parameters than can be justified by the data.In a mathematical sense, these parameters represent the degree of polynomial. The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e., the noise) as if that variation represented the underlying model structure.

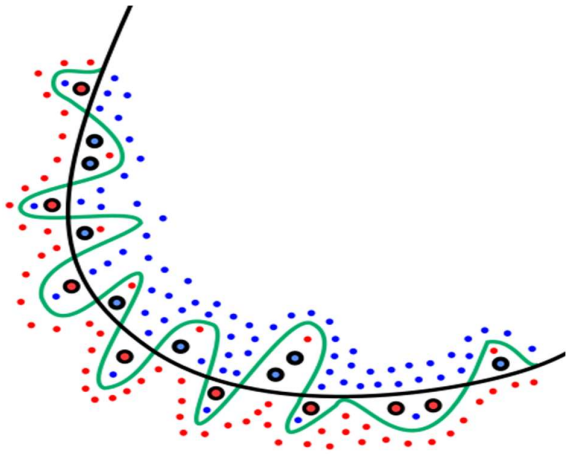


Fig4: Illustration 1 of Overfitting

The green line represents an overfitted model and the black line represents a regularised model. While the green line best follows the training data, it is too dependent on that data and it is likely to have a higher error rate on new unseen data illustrated by black-outlined dots, compared to the black line.

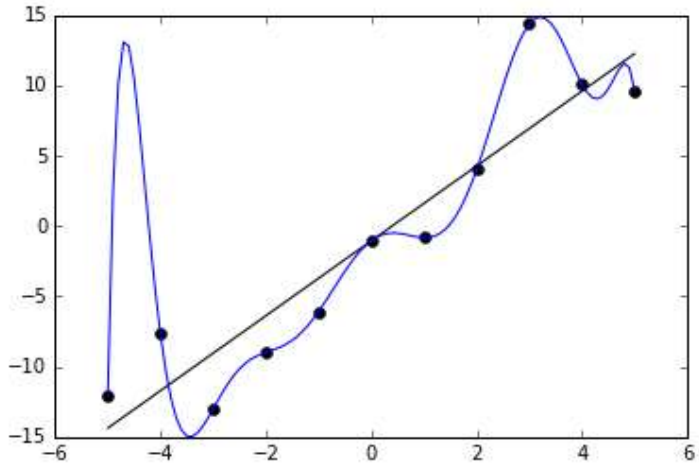


Fig5: Illustration 2 of Overfitting

Noisy (roughly linear) data is fitted to a linear function and a polynomial function. Although the polynomial function is a perfect fit, the linear function can be expected to generalise better: if the two functions were used to extrapolate beyond the fitted data, the linear function should make better predictions.

Some prominent methods to resolve overfitting:

1. Reduce model complexity.
2. Get more data.
3. Use early stopping techniques
4. Use L1/L2 regularisation or batch normalisation techniques.
5. Use validation data to stop when model starts overfitting
6. Use adaptive learning rate techniques.

In summary overfitting is 2nd case amongst the 4 different cases mentioned below:-

Underfitting – Validation and training error high

Overfitting – Validation error is high, training error low

Good fit – Validation error low, slightly higher than the training error

Unknown fit - Validation error low, training error 'high'.

And we have discussed above some of the illustrations of overfitting and the methods to resolve it.

From the next section onwards the observations for question 1 have been presented.

Note: When we consider 700 train samples, those must include 100 train samples considered for 100 train samples case, those 100 train samples must in turn have 50 train samples considered for

the case of 50 train samples case which in turn must have the 10 train samples considered for the 10 train samples case.

Observations for 700 samples:

Plot for train data which is 700 train sample case.

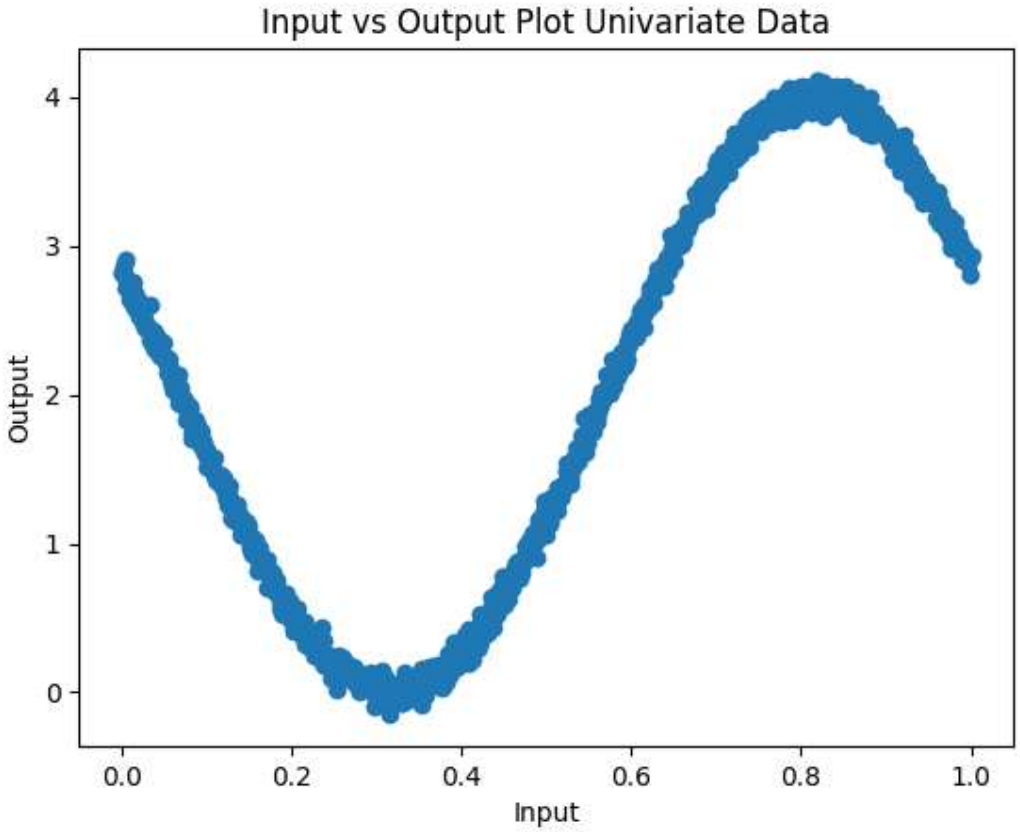


Fig6

- The values of x should lie between 0 and 1 (they being included) as mentioned earlier but that may not be the case as those values might have gone to the test dataset.
- The y values should lie between -0.14576 and 4.1118 as mentioned earlier may or may not be present here as they might have gone to the test dataset which will be shown next. But despite this overall we can say the range and the domain of the train data is almost the same as earlier.
- The number of train data points is 700 which is the case where we consider the whole training data (whole 70% of 1001).

Plot of the test data for 700 train samples case.

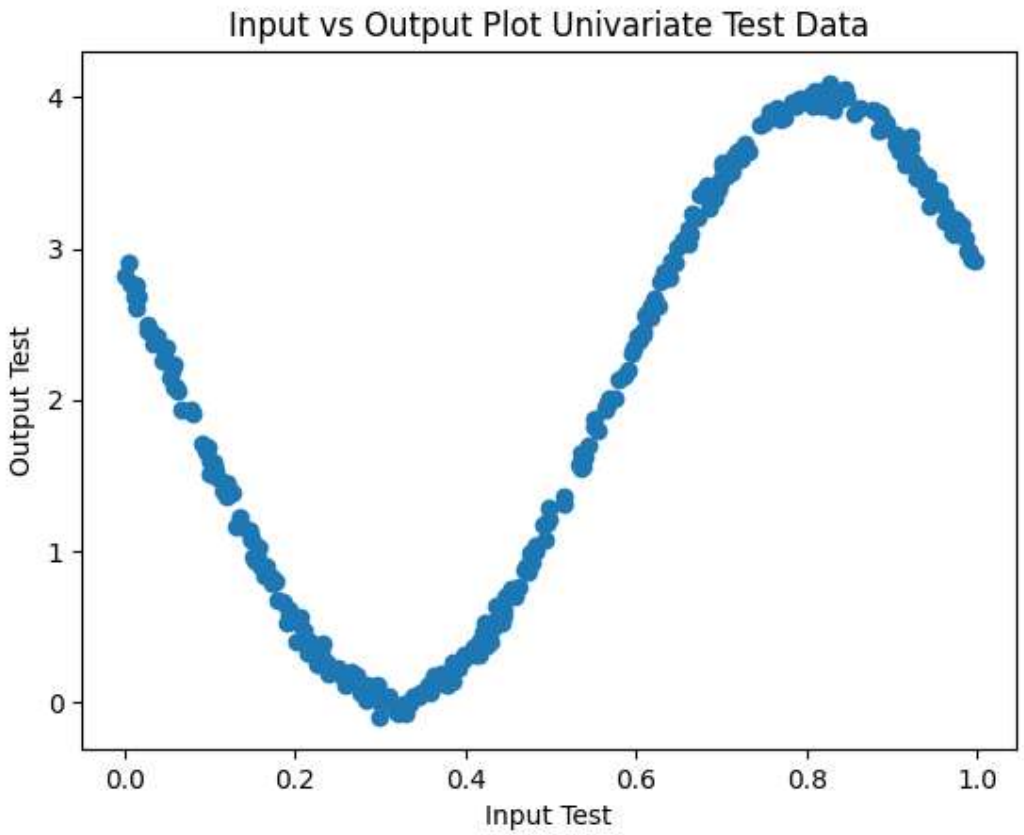


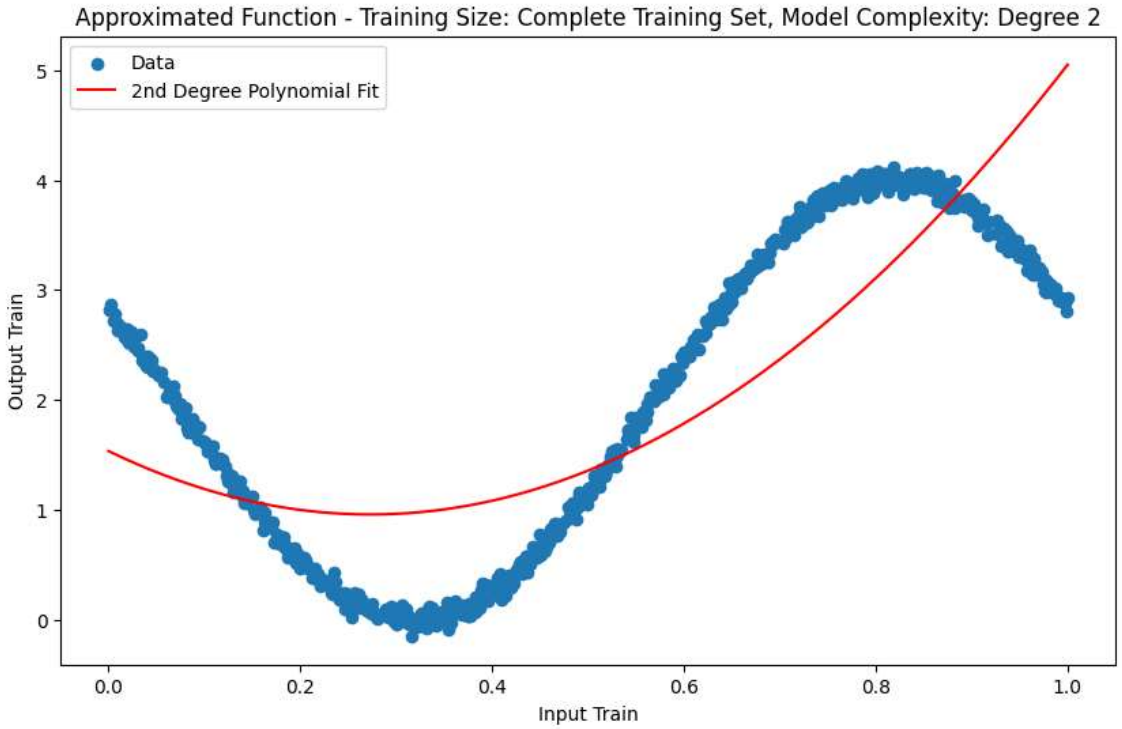
Fig7

- The values of x roughly lie between 0 and 1 (they being included).
- The y values roughly lie between -0.14576 and 4.1118.
- The number of test data points is 300 (whole 30% of 1001).

We can observe from the train and test data that a linear or quadratic polynomial won't be a good fit and we need at least a cubic polynomial or more for a good fit.

Results:

1) **For degree/model complexity being 2**



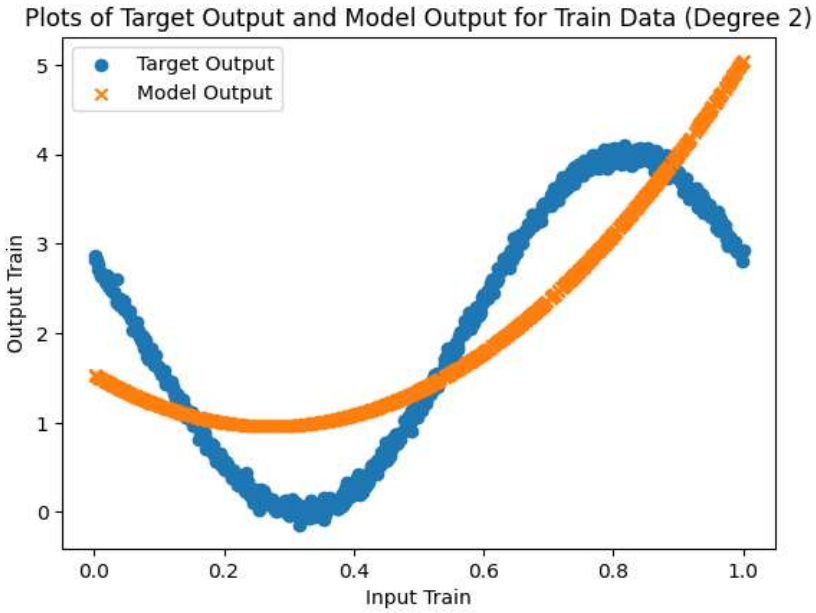


Fig8 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 2)

Fig9 **Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 2)**

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 2:	1.5354873486201916
w1 value of degree 2:	-4.222844862810273
w2 value of degree 2:	7.7359767656423255
MSE for degree 2 training data:	0.682887427095117
MSE Percent for degree 2 train data:	34.03295944167583
MSE for degree 2 test data:	0.6661028561078643
MSE Percent for degree 2 test data:	33.30306882635563

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself just to give a peek of the train and test MSE. The explanation will be given later on.

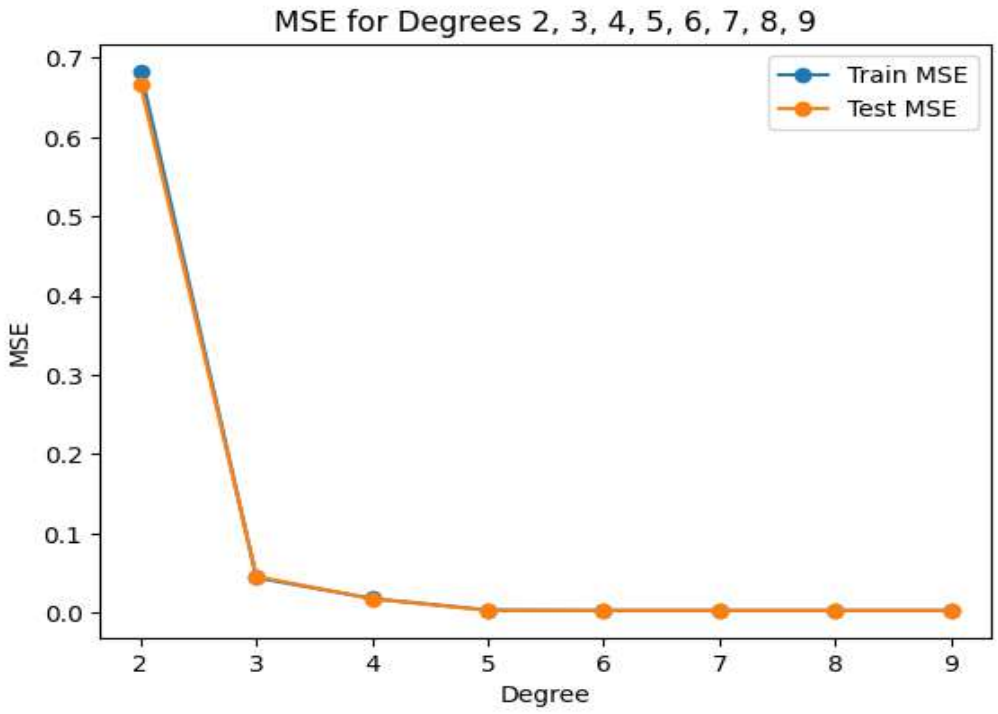


Fig10 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

2) For degree/model complexity being 3

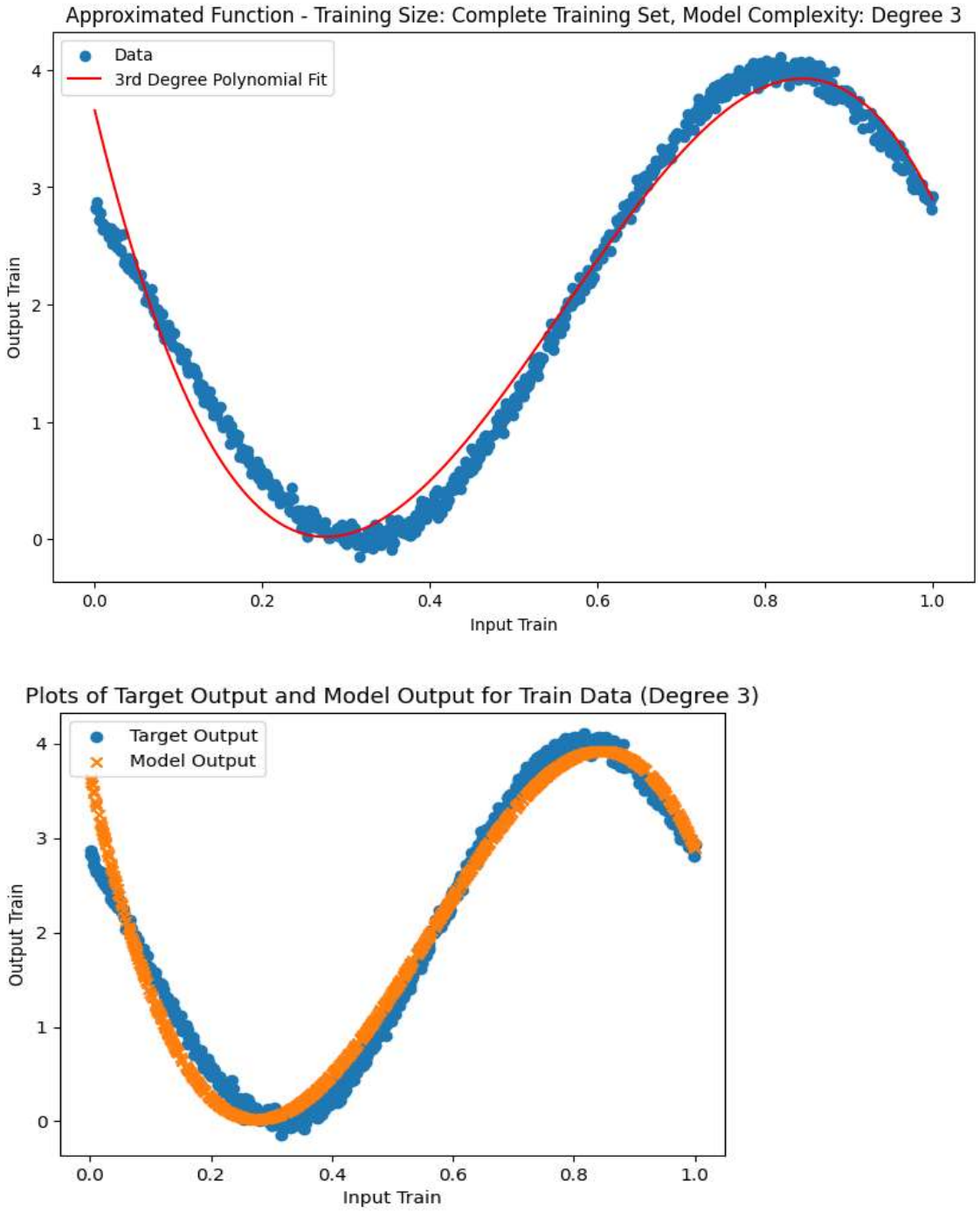


Fig11 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 3)

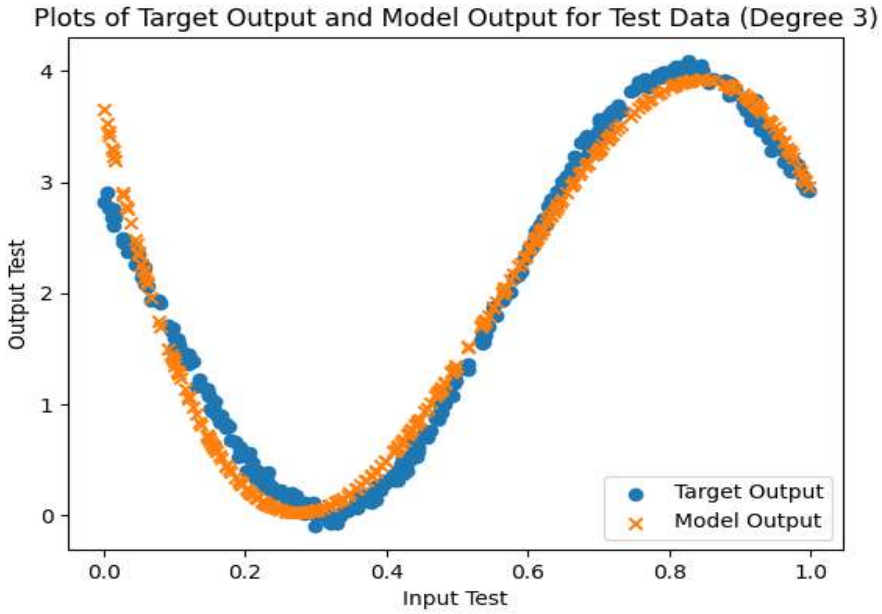


Fig12 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 3)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 3:	3.653729758705805
w1 value of degree 3:	-29.573006318041244
w2 value of degree 3:	71.1367721357019
w3 value of degree 3:	-42.32032795065878
MSE for degree 3 training data:	0.044305192988328405
MSE Percent for degree 3 train data:	2.208031333131251
MSE for degree 3 test data:	0.04581063043746698
MSE Percent for degree 3 test data:	2.290388885812934

We observe that **degree 3 is a better fit than degree 2** and **both train and test MSE have reduced significantly (steep decline)** for instance from **degree 2 to 3 train error has reduced from 0.68 to 0.044** and **for test data test error has reduced from 0.666 to 0.0458.**

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

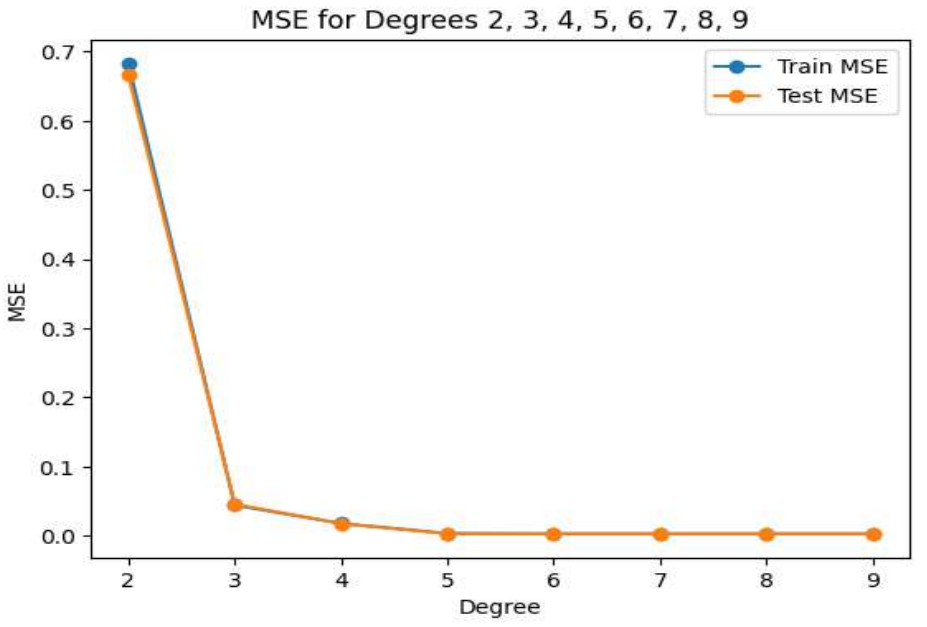
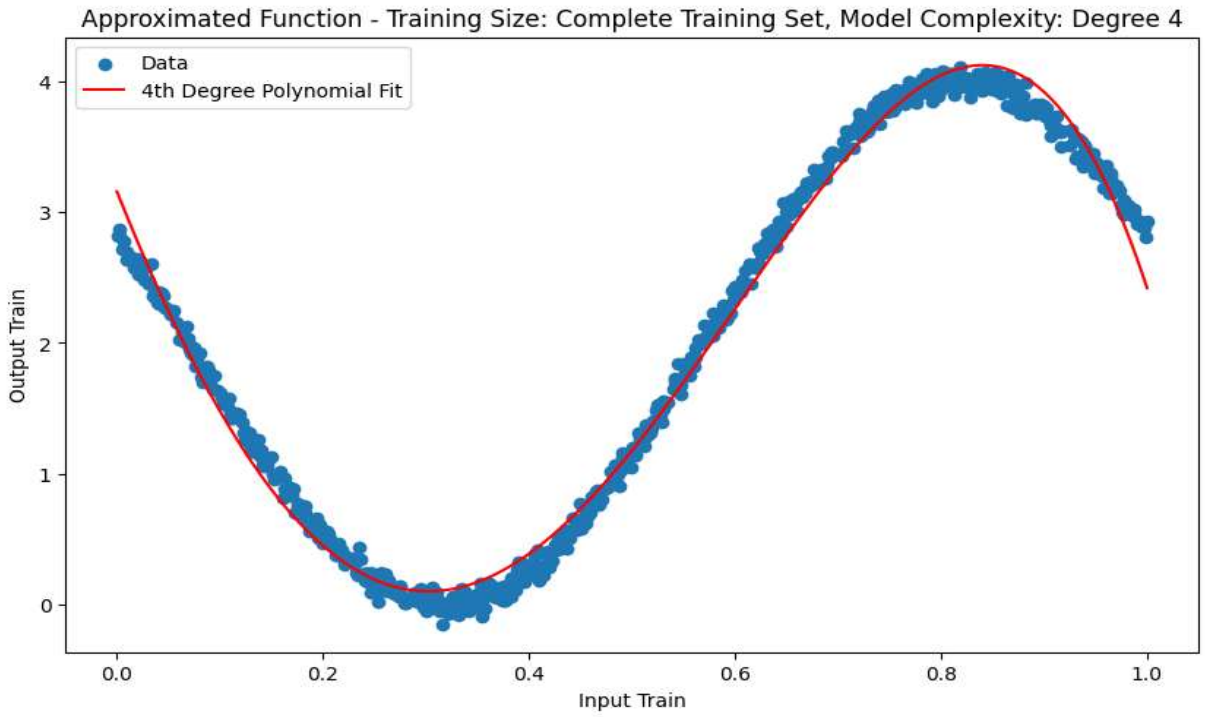


Fig13 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

3) For degree/model complexity being 4



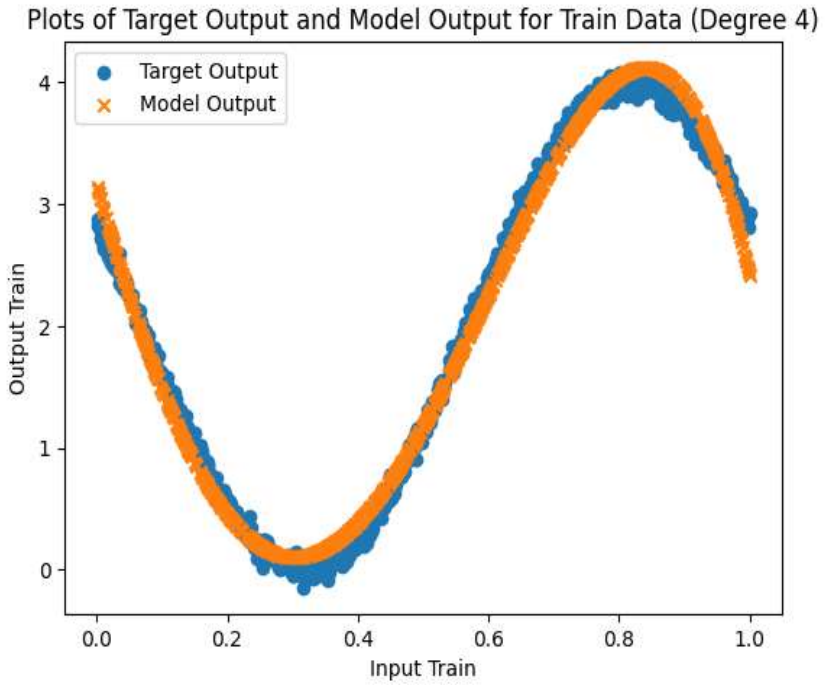


Fig14 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 4)

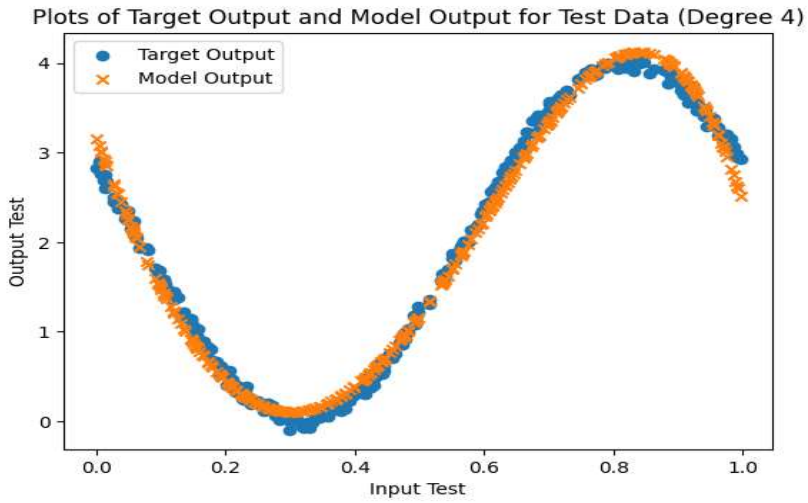


Fig15 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 4)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 4:	3.156234929109081
w1 value of degree 4:	-19.68163391546991
w2 value of degree 4:	26.9889207423628
w3 value of degree 4:	25.921472963749064
w4 value of degree 4:	-33.96457383856684
MSE for degree 4 training data:	0.017907589999535935
MSE Percent for degree 4 training data:	0.8924579073667412
MSE for degree 4 test data:	0.01756283330533223
MSE Percent for degree 4 test data:	0.8780869815975926

We observe that: -

- Degree 4 is a better fit than degree 3.
- Both train and test MSE have reduced from degree 3 to 4 but not as drastically as they did from degree 2 to 3.
- Degree 3 to 4 train error has reduced from 0.044 to 0.0179.
- For test data, test error has reduced from 0.0458 to 0.0175.

Moving onto 4 degree from 3 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 2 to 3 which is the case as observed from the values of train and test MSE.

This indicates the model has probably become a decent fit for the data and further increase in degrees may not be that beneficial as it may lead to:

- **Overfitting (model may not be able to generalise).**
- **More complexity**.
- **Requirement of more training samples** (the thumb rule mentioned earlier).
- **Very less reduction in train and test MSE values**.
- **Huge coefficient values which in other words means incurring the overhead/cost of regularisation in case of the overfitting.**

Whether these things actually occur in case of train samples being 700 (a decent amount even for 9 degree polynomials) is becoming evident as we are proceeding ahead and obtaining results for higher degrees.

A point to observe is that the values of our parameters (w0, w1, w2, w3) are not that extreme, hence apparently there is no need of regularisation as of now.

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

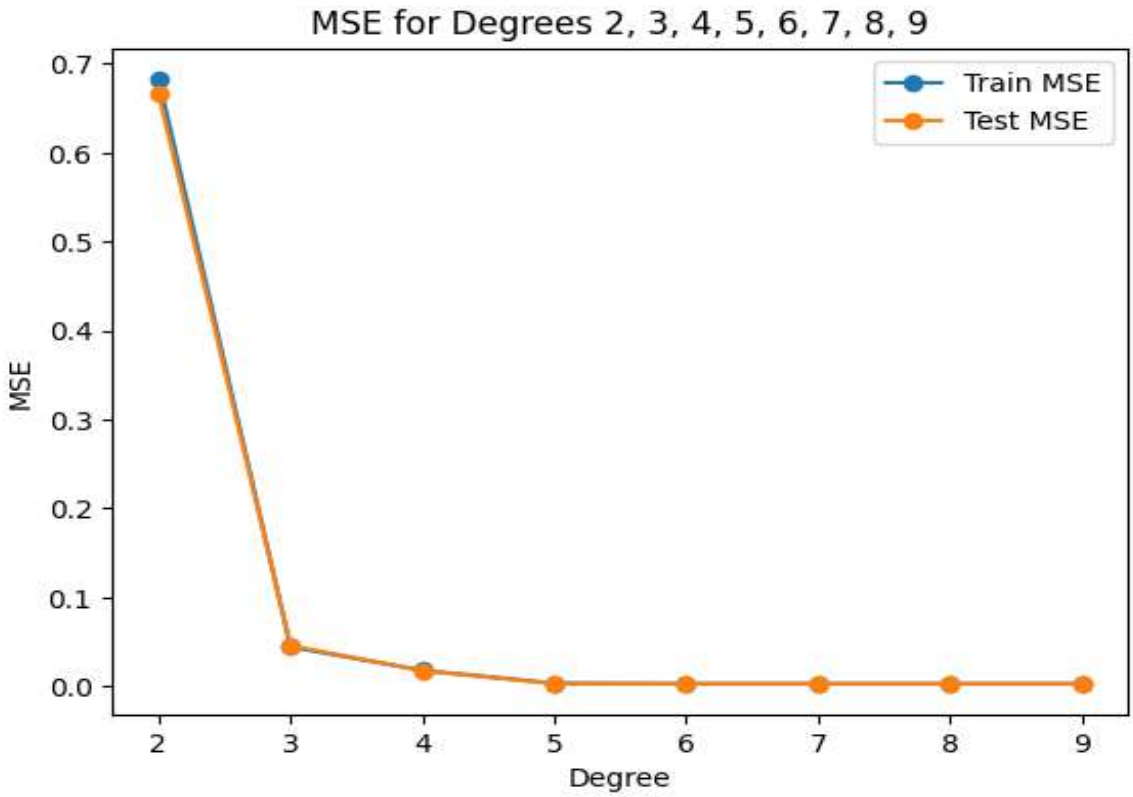


Fig16 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

4) For degree/model complexity being 5

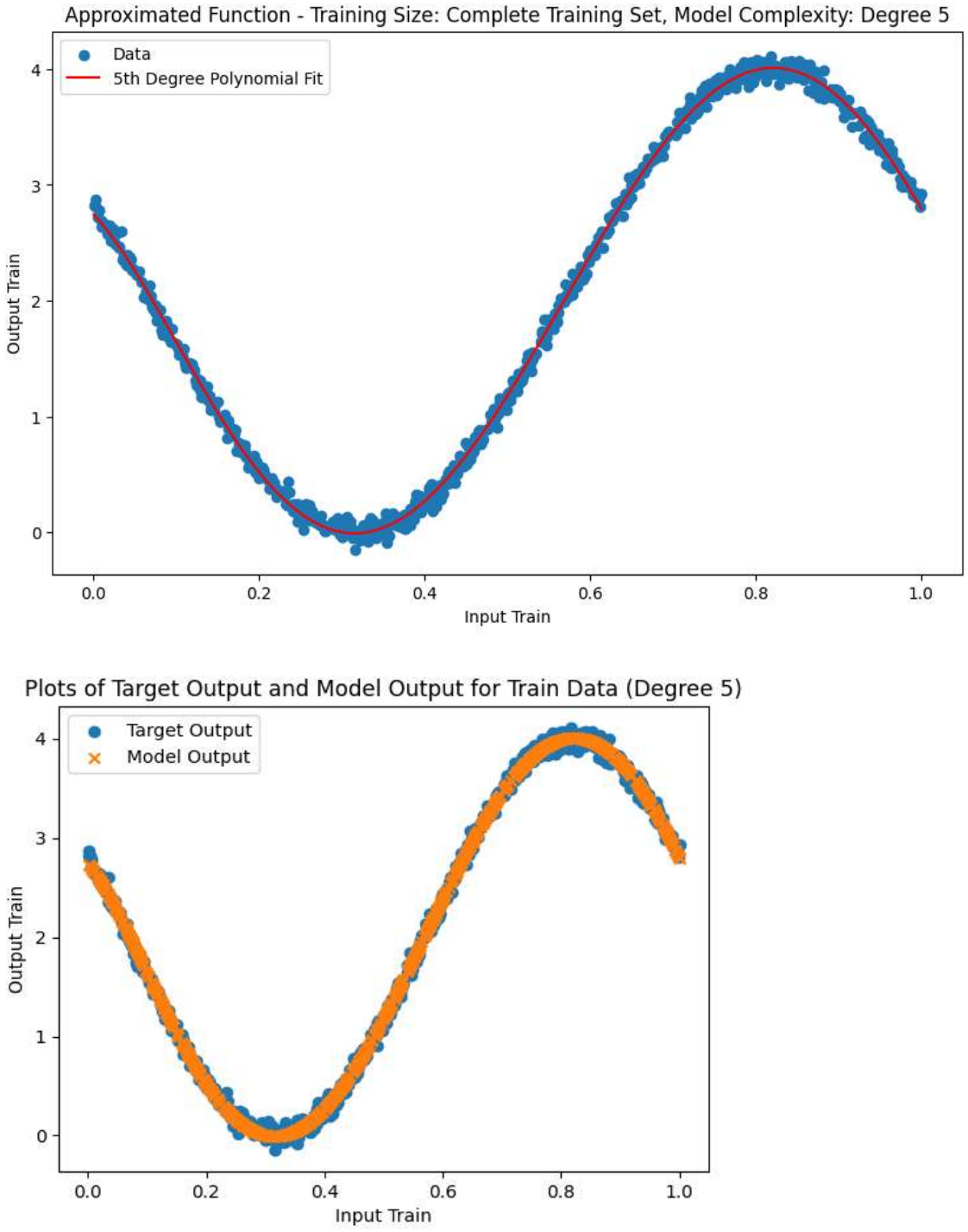


Fig17 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 5)

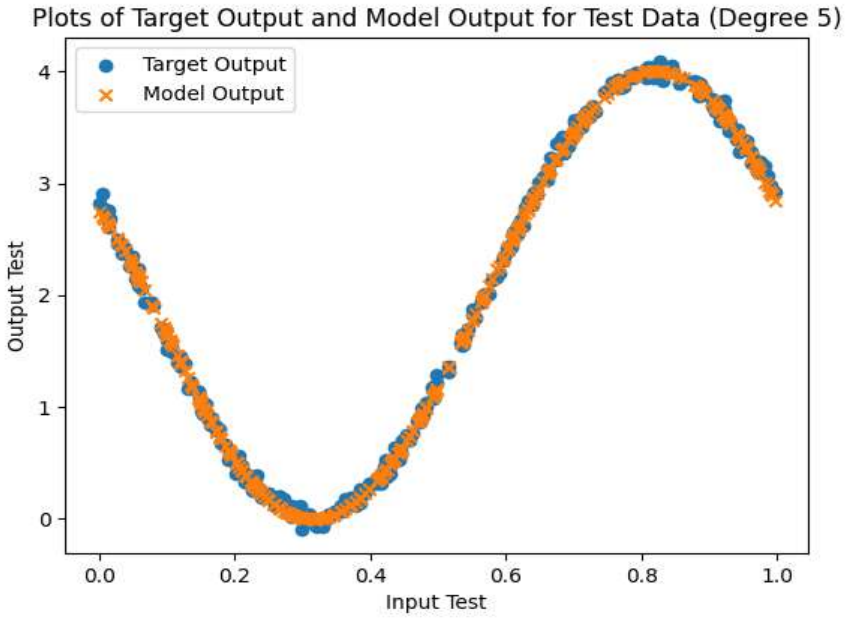


Fig18 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 5)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 5:	2.7426771193124964
w1 value of degree 5:	-7.429931215113797
w2 value of degree 5:	-57.98749434131966
w3 value of degree 5:	251.195728402259
w4 value of degree 5:	-286.27087431590576
w5 value of degree 5:	100.54777424404817
MSE for degree 5 training data:	0.003445616753256478
MSE Percent for degree 5 train data:	0.171718691196233
MSE for degree 5 test data:	0.0029734692829770258
MSE Percent for degree 5 test data:	0.14866420594960283

We observe that: -

- Degree 5 is a better fit than degree 4.
- Both train and test MSE have reduced from degree 4 to 5 but not as drastically as they did from degree 3 to 4 and the reduction is extremely minute.
- Degree 4 to 5 train error has reduced from 0.0179 to 0.0034.
- For test data, test error has reduced from 0.0175 to 0.0029 .

But what we now observe is that **values of our parameters have increased significantly from degree 4 to 5** and the **error has declined significantly less** which means **we had most probably obtained a perfect fit** at degree 4 and **we should not have moved ahead** as we will be invoking the need of regularisation because of huge parameter values without gaining any benefit in terms of train and test MSE values.

But since **we have a decent number of training samples, our test MSE might not increase (preventing the condition of overfitting) but values of our coefficients might increase drastically at any point and the model might lose its generalisation ability.**

Moving onto 5 degree from 4 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 3 to 4 which is the case as observed from the values of train and test MSE.

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

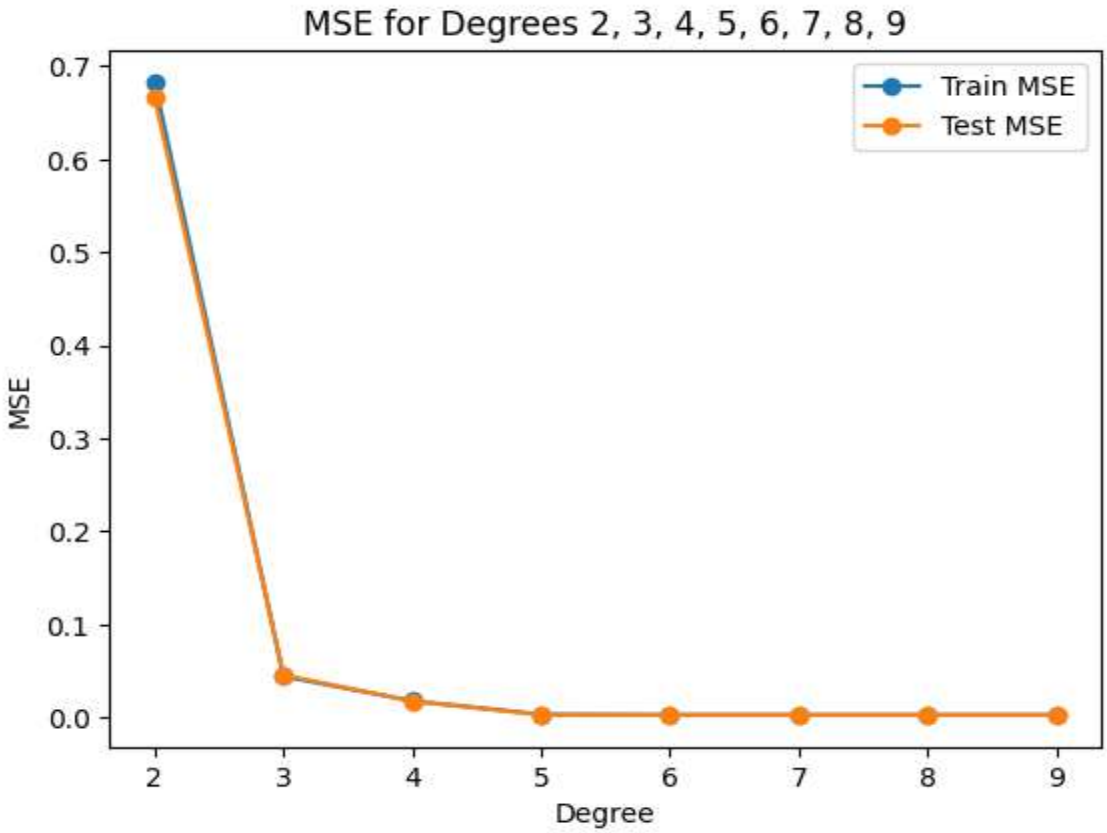


Fig19 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

5) For degree/model complexity being 6

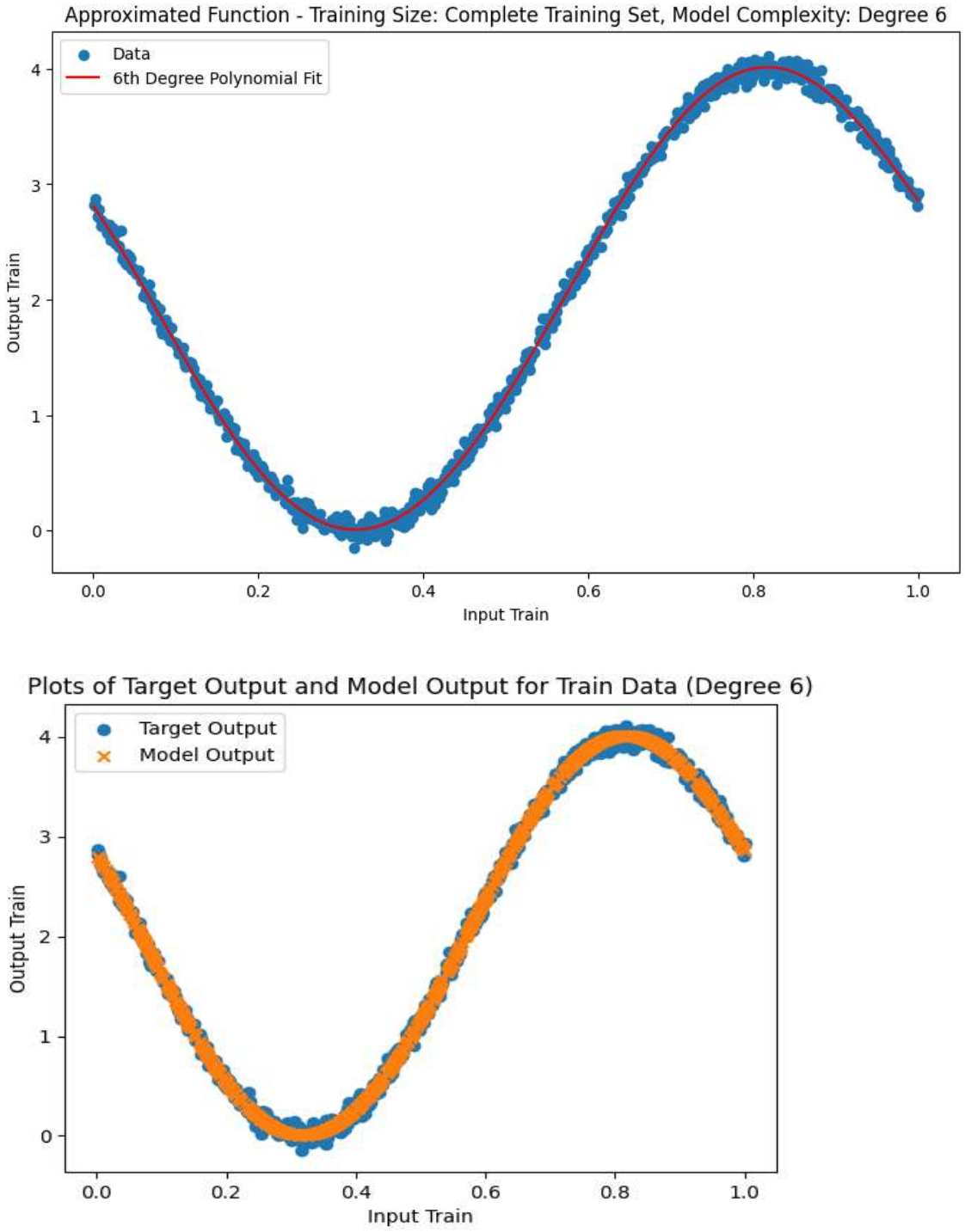


Fig20 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 6)

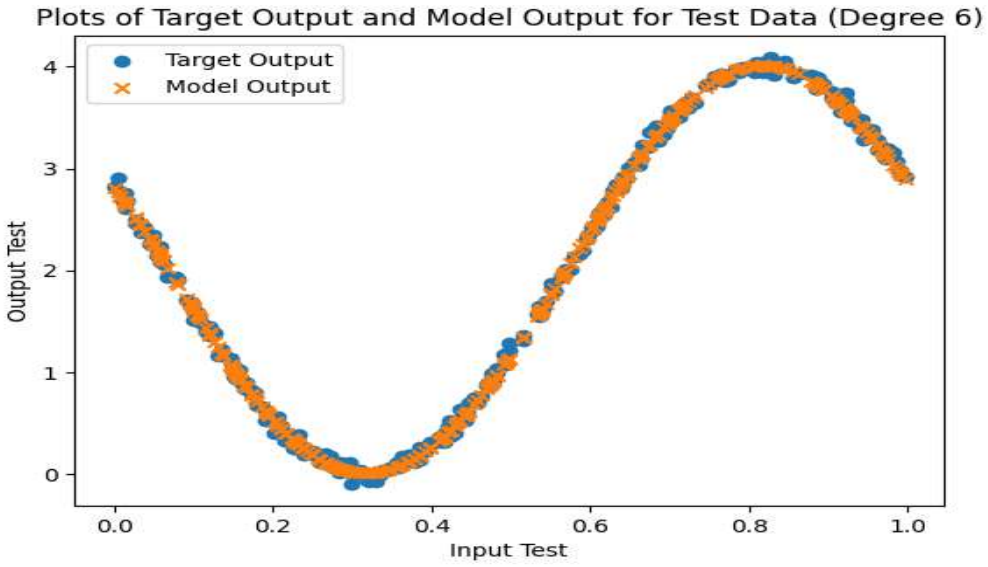


Fig21 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 6)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 6:	2.8081894524225124
w1 value of degree 6:	-10.137709604080388
w2 value of degree 6:	-31.066456656216793
w3 value of degree 6:	143.82026010538317
w4 value of degree 6:	-85.27300783960037
w5 value of degree 6:	-76.13632807851525
w6 value of degree 6:	58.8432274463043
MSE for degree 6 training data:	0.0031332552601137406
MSE Percent for degree 6 training data:	0.15615157778123176
MSE for degree 6 test data:	0.002799846299467151
MSE Percent for degree 6 test data:	0.13998359736693938

We observe that:-

- Degree 6 is a better fit than degree 5.
- Both train and test MSE have reduced from degree 5 to 6 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.
- Degree 5 to 6 train error has reduced from 0.0034 to 0.0031.
- For test data, test error has reduced from 0.0029 to 0.0027 .

Moving onto 6 degree from 5 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 4 to 5 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have not increased significantly from degree 5 to 6** and the **error has declined but it is significantly less** which means **we are making our model more complex, it is losing its generalisation ability and the coefficients might soon be large at any point so we should select some previous degree model as a suitable one.**

That being said, our test MSE is still low since we have a decent number of training samples but the model is still losing its generalisation ability.

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

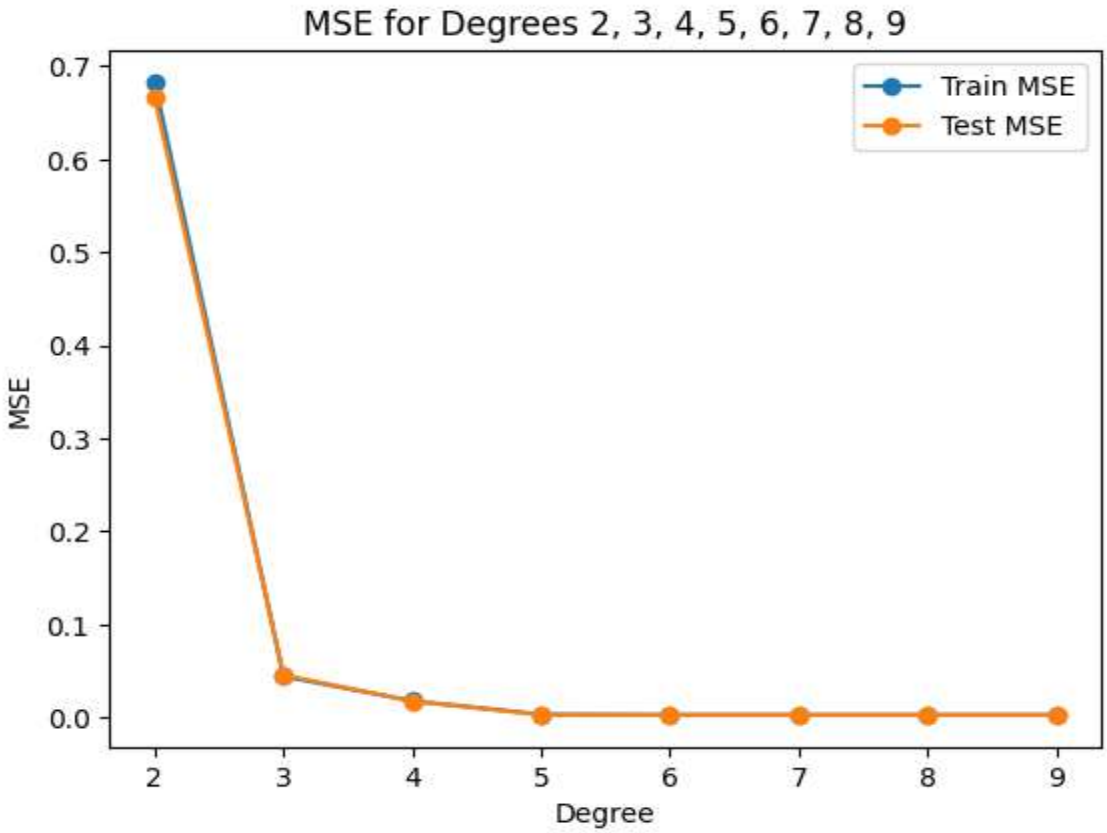
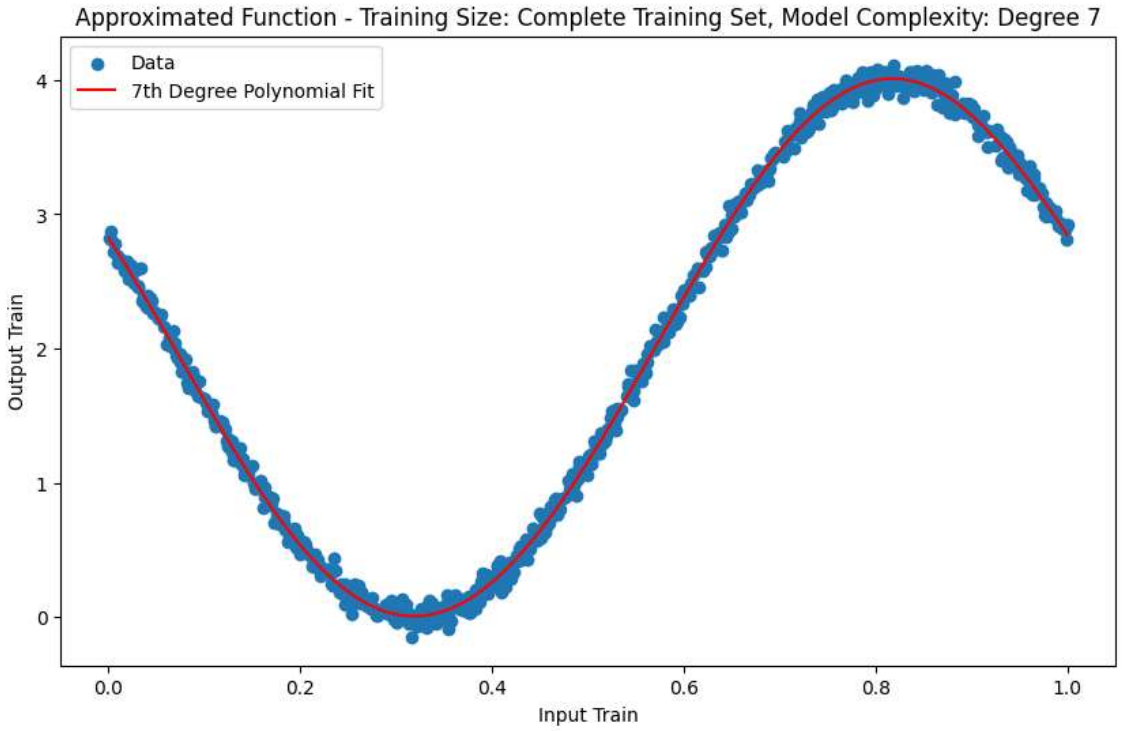


Fig22 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which may result in huge values of parameters.

And as we see that is the case for results of degrees we see ahead.

6) For degree/model complexity being 7



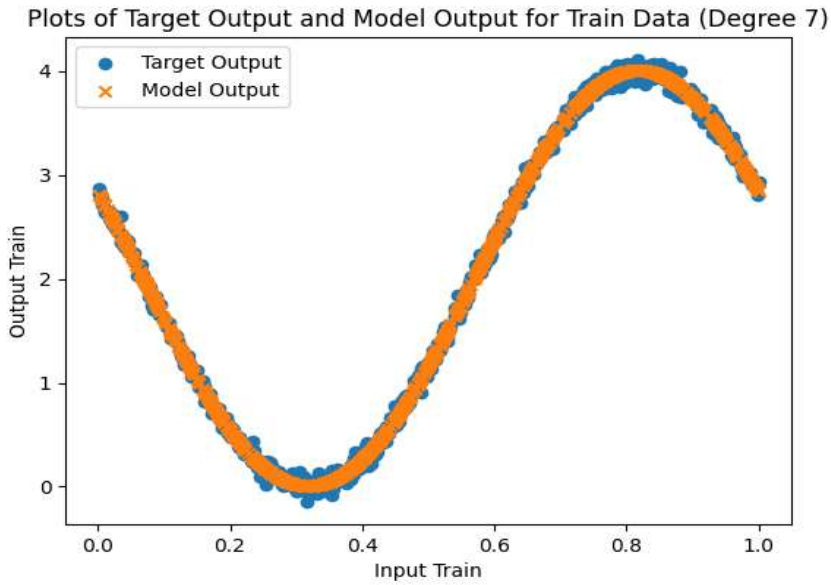


Fig23 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 7)

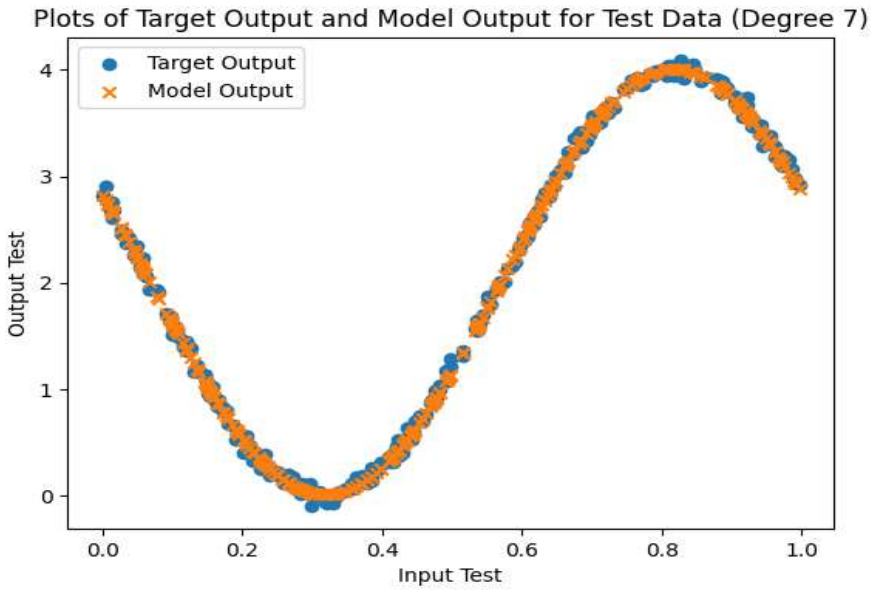


Fig24 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 7)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 7:	2.82221696434529
w1 value of degree 7:	-10.91075020392553
w2 value of degree 7:	-20.677870997763137
w3 value of degree 7:	86.30913828049029
w4 value of degree 7:	72.41964228181678
w5 value of degree 7:	-302.65620453091196
w6 value of degree 7:	222.10226263358254
w7 value of degree 7:	-46.56292102374505
MSE for degree 7 training data:	0.003121031410703402
MSE Percent for degree 7 training data:	0.1555423796107912
MSE for degree 7 test data:	0.0027693475931665494
MSE Percent for degree 7 test data:	0.1384587570127355

We observe that:-

- Degree 7 is a better fit than degree 6.
- Both train and test MSE have reduced from degree 6 to 7 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.
- Degree 6 to 7 train error has reduced from 0.00313 to 0.00312.
- For test data, test error has reduced from 0.00279 to 0.00276 .

Moving onto 7 degree from 6 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 5 to 6 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 6 to 7** and the **MSE values have stayed the same so we should have stopped earlier/ at degree 5 or 4 as discussed earlier (since degree 5 seems to have less generalisation capacity as compared to degree 4).**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and train MSE and the model most probably has now lost its generalisation ability.

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

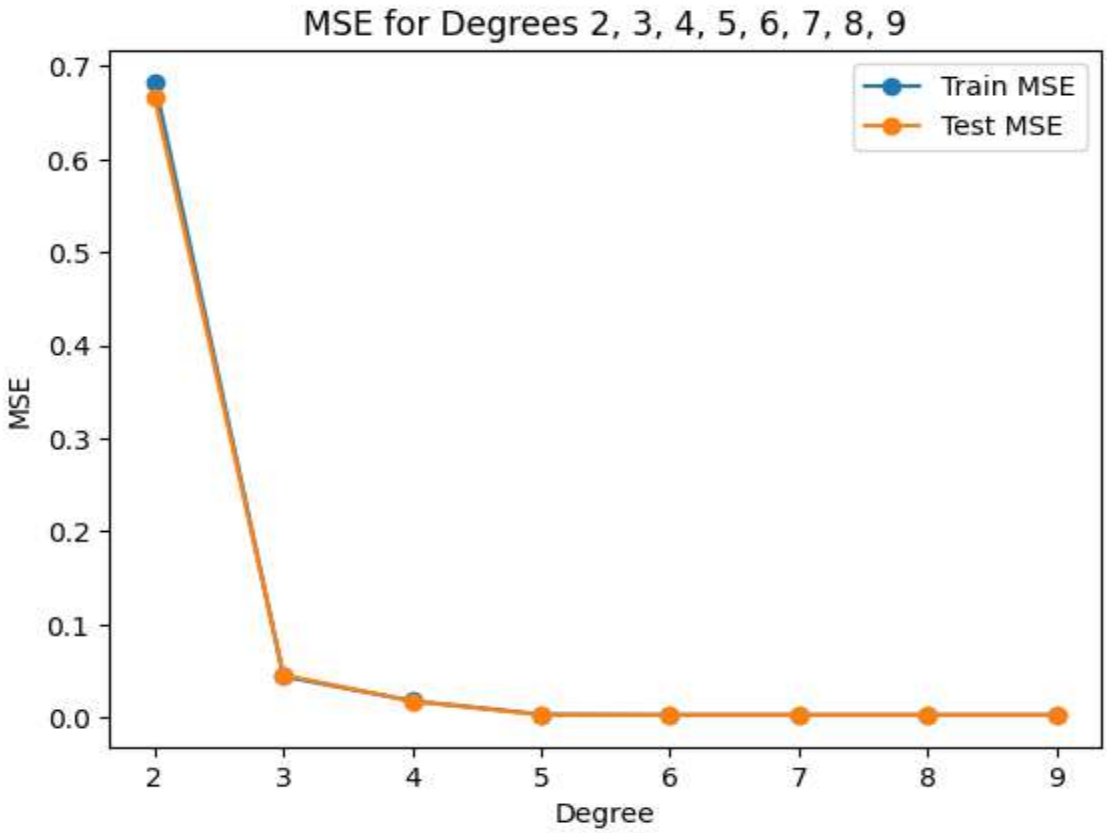


Fig25 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing complexity of the model which may result in huge values of parameters, but since we have decent number of training samples so in theory we should be able to observe no increase in test MSE.

And as we see that is the case for results of degrees we see ahead.

7) **For degree/model complexity being 8**

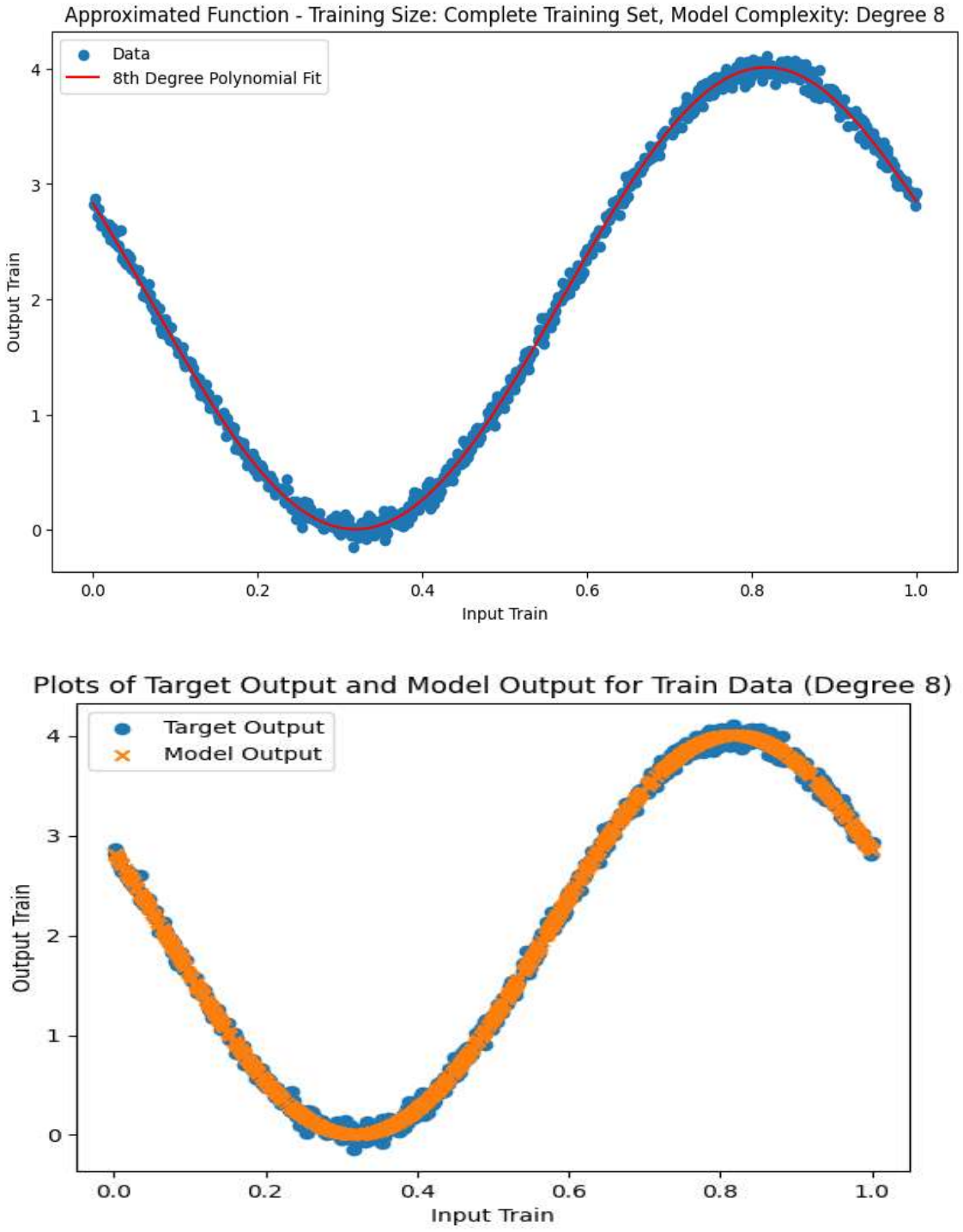


Fig26 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 8)

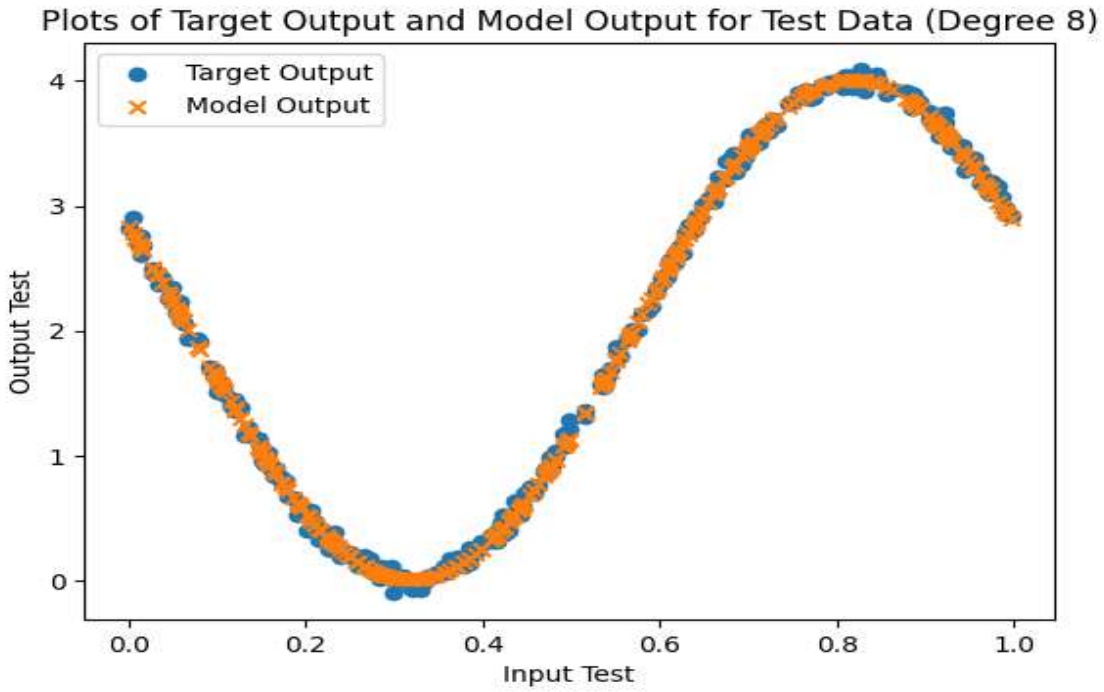


Fig27 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 8)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 8:	2.8275716756888887
w1 value of degree 8:	-11.290158580855078
w2 value of degree 8:	-14.070126675177903
w3 value of degree 8:	38.022785124066104
w4 value of degree 8:	253.0292184749469
w5 value of degree 8:	-677.6147922773089
w6 value of degree 8:	658.9219235305609
w7 value of degree 8:	-313.69464129646985
w8 value of degree 8:	66.71856926141663
MSE for degree 8 training data:	0.0031194284838429467
MSE Percent for degree 8 training data:	0.15546249478253787
MSE for degree 8 test data:	0.0027608397999365443
MSE Percent for degree 8 test data:	0.13803339384111535

We observe that:-

- Degree 8 is a better fit than degree 7.
- Both train and test MSE have reduced from degree 7 to 8 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.
- Degree 7 to 8 train error has reduced from 0.00312 to 0.00311.
- For test data, test error has not changed from 0.00276.

Moving onto 8 degree from 7 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 6 to 7 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 7 to 8** and the **MSE values have stayed the same.**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and training MSE and the model most probably has now completely lost its generalisation ability.

We shall see the train and test MSE graphs afterwards for all the degrees for 700 train samples and 300 test samples but since we have mentioned the MSE values so it makes sense to see them here itself. The explanation will be given later on.

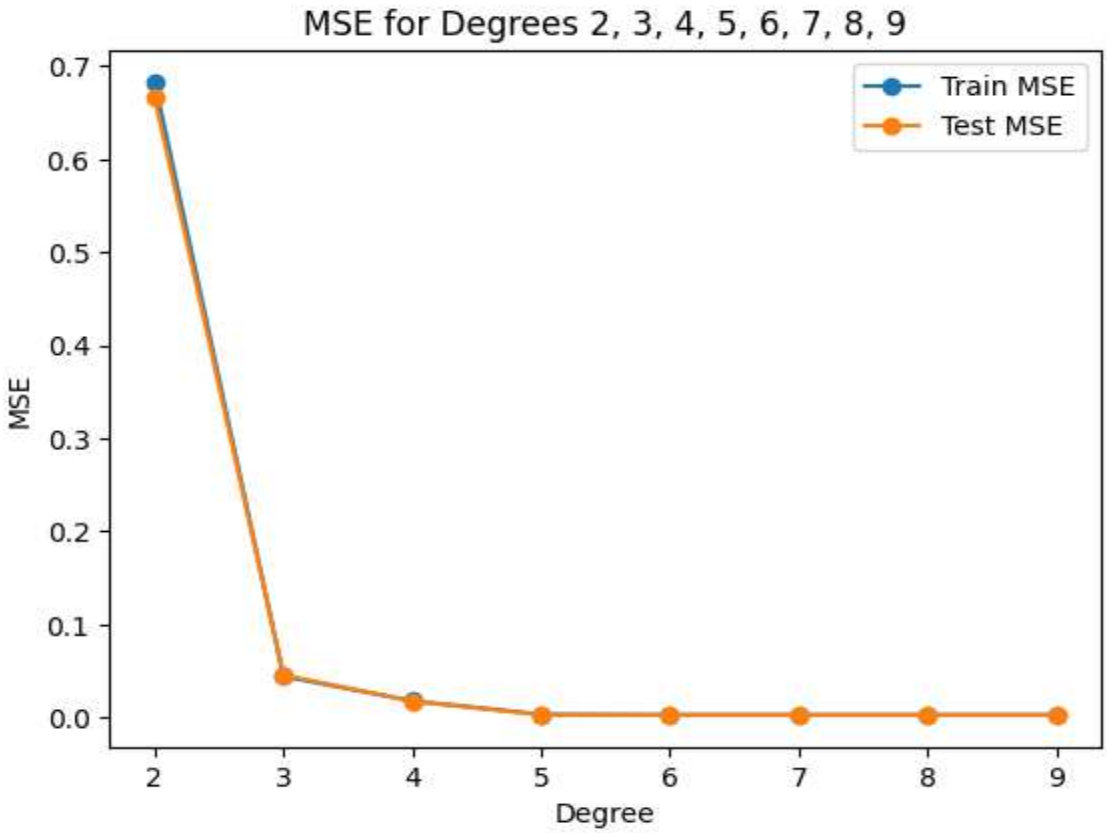
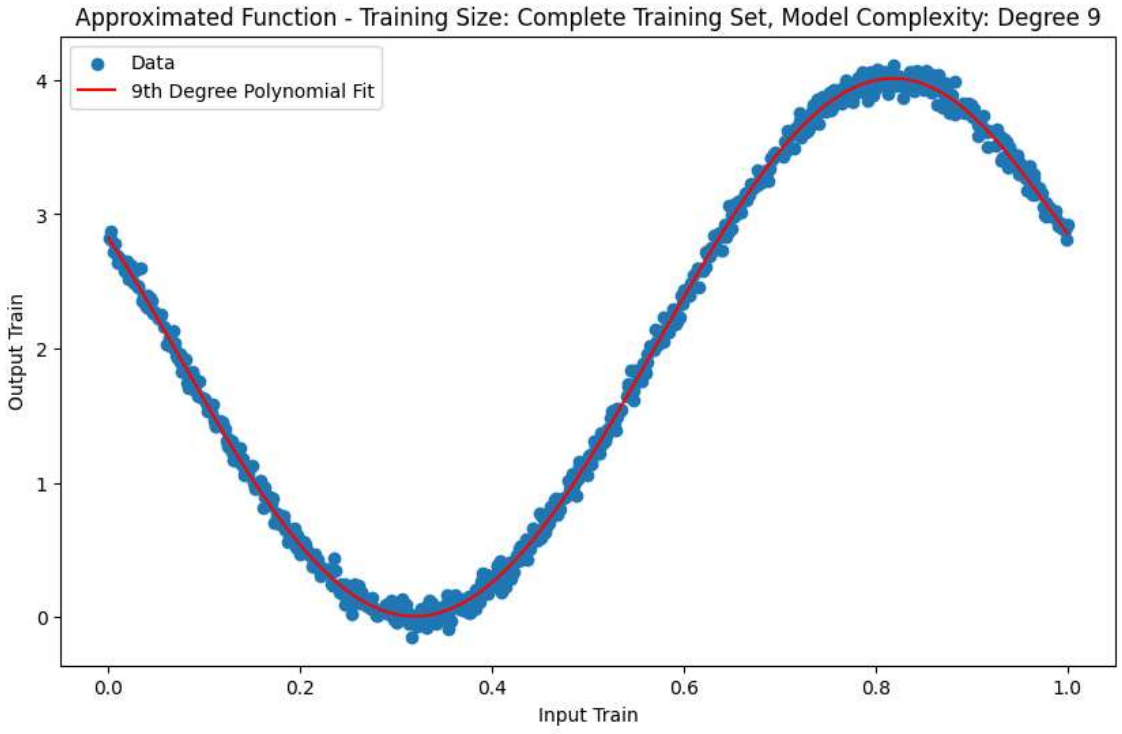


Fig28 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing complexity of the model which may result in huge values of parameters, but since we have decent number of training samples so in theory we should be able to observe no increase in test MSE.

And as we see that is the case for results of degrees we see ahead.

8) For degree/model complexity being 9



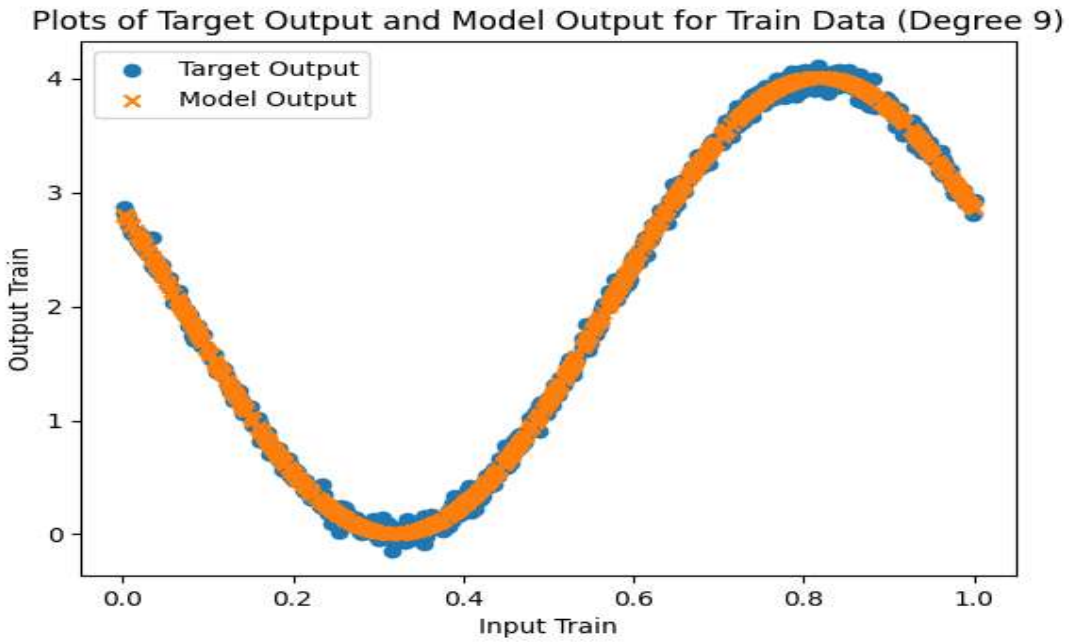


Fig29 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 9)

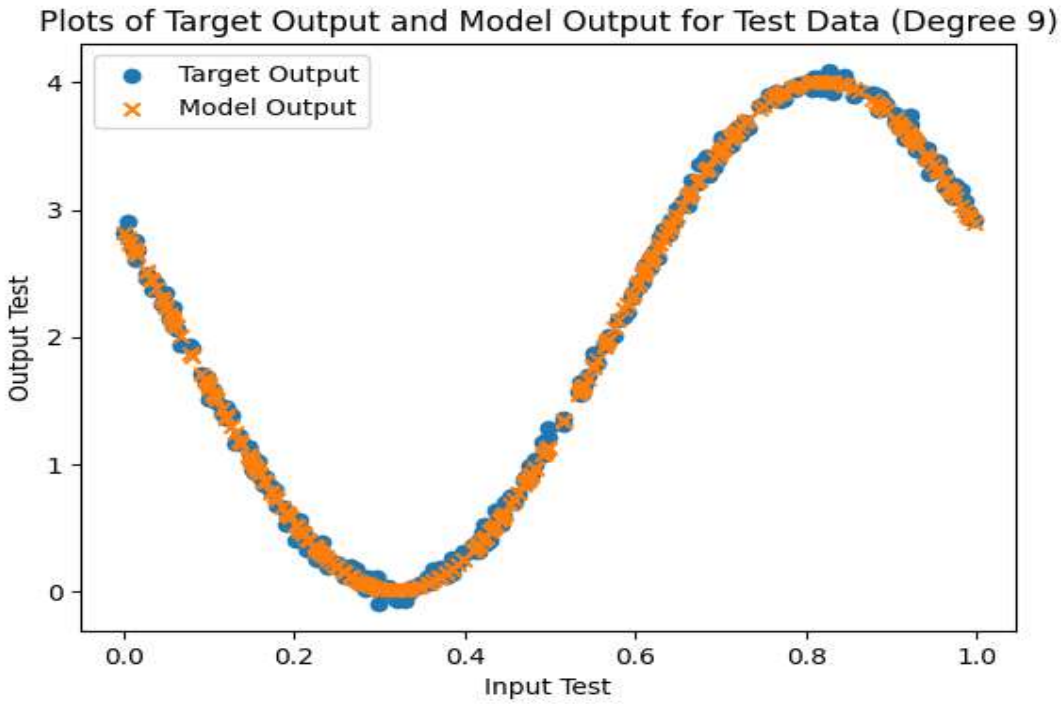


Fig30 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 9)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 9:	2.82005592193318
w1 value of degree 9:	-10.608127367657474
w2 value of degree 9:	-28.999180385665795
w3 value of degree 9:	176.6868859096976
w4 value of degree 9:	-420.0512625708943
w5 value of degree 9:	1199.7296789396496
w6 value of degree 9:	-2459.0428296942555
w7 value of degree 9:	2730.9199754823057
w8 value of degree 9:	-1546.030626058564
w9 value of degree 9:	357.43236402791626
MSE for degree 9 training data:	0.003116534707622286
MSE Percent for degree 9 training data:	0.15531827808613455
MSE for degree 9 test data:	0.00277467618376161
MSE Percent for degree 9 test data:	0.13872516995137935

We observe that:-

- Degree 9 is a better fit than degree 8.
- Both train and test MSE have reduced from degree 8 to 9 but not as drastically as they did from degree 7 to 8 and the reduction is extremely minute.
- Degree 8 to 9 train error has stayed the same.
- For test data, test error has stayed the same.

Moving onto 9 degree from 8 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 7 to 8 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 8 to 9** and the **MSE values have stayed the same.**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and training MSE and the model has now completely lost its generalisation ability.

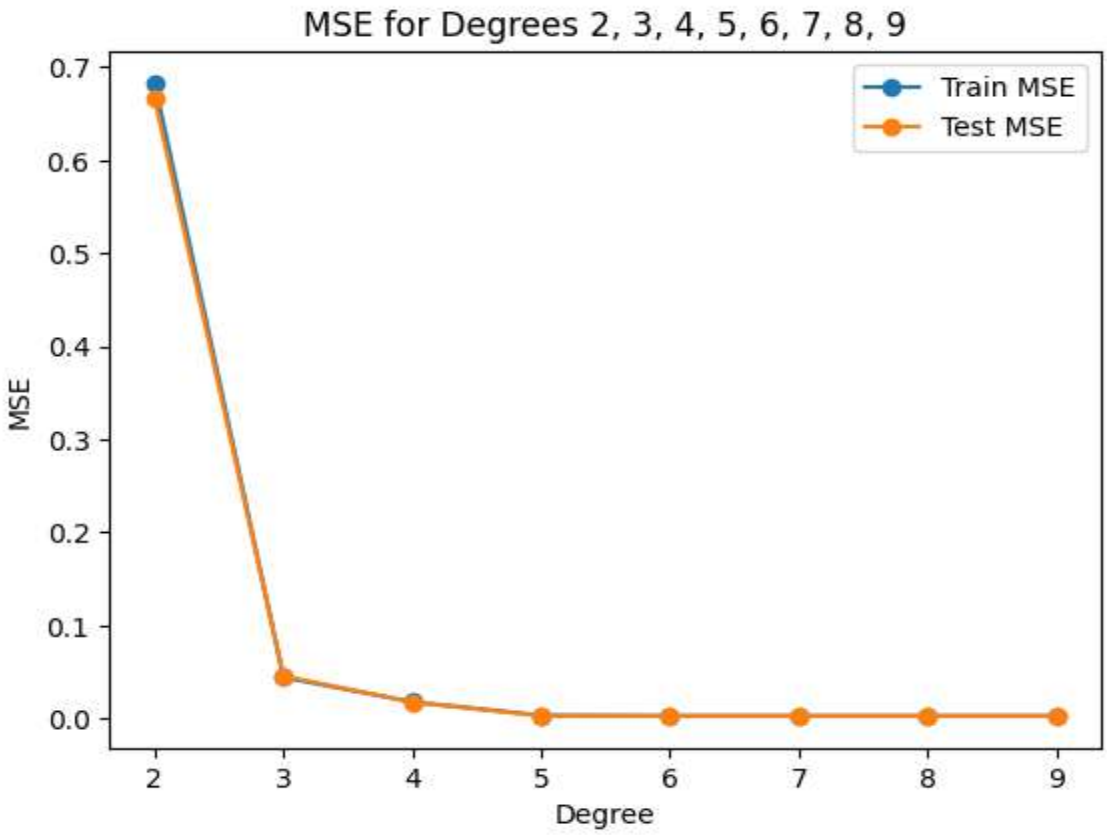


Fig31 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

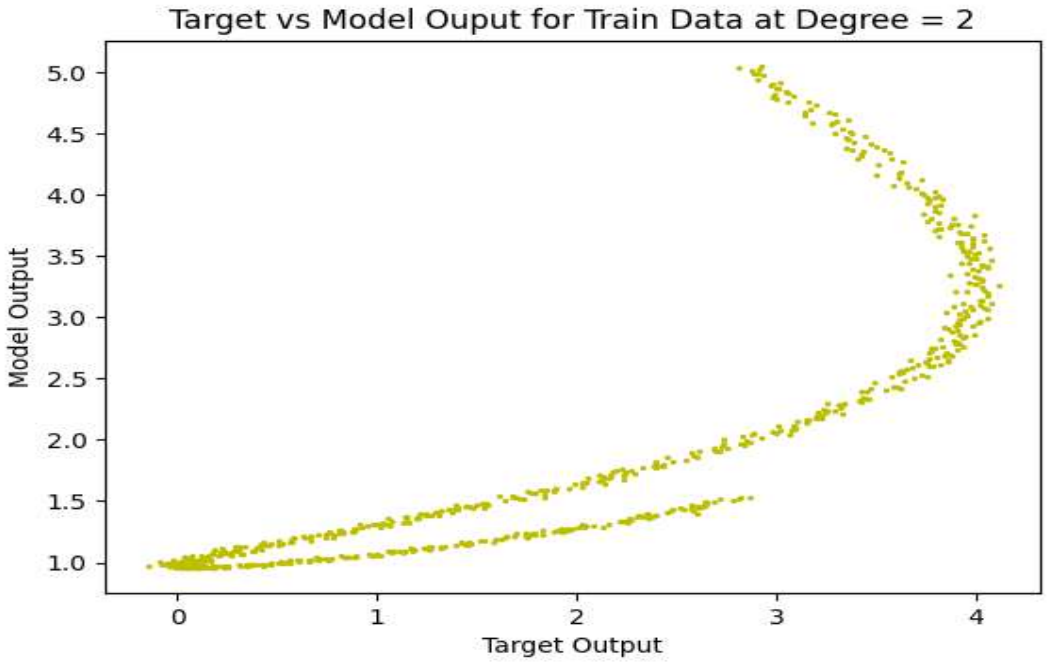
We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing complexity of the model which may result in huge values of parameters, but since we have decent number of training samples so in theory we should be able to observe no increase in test MSE as compared to the train MSE.

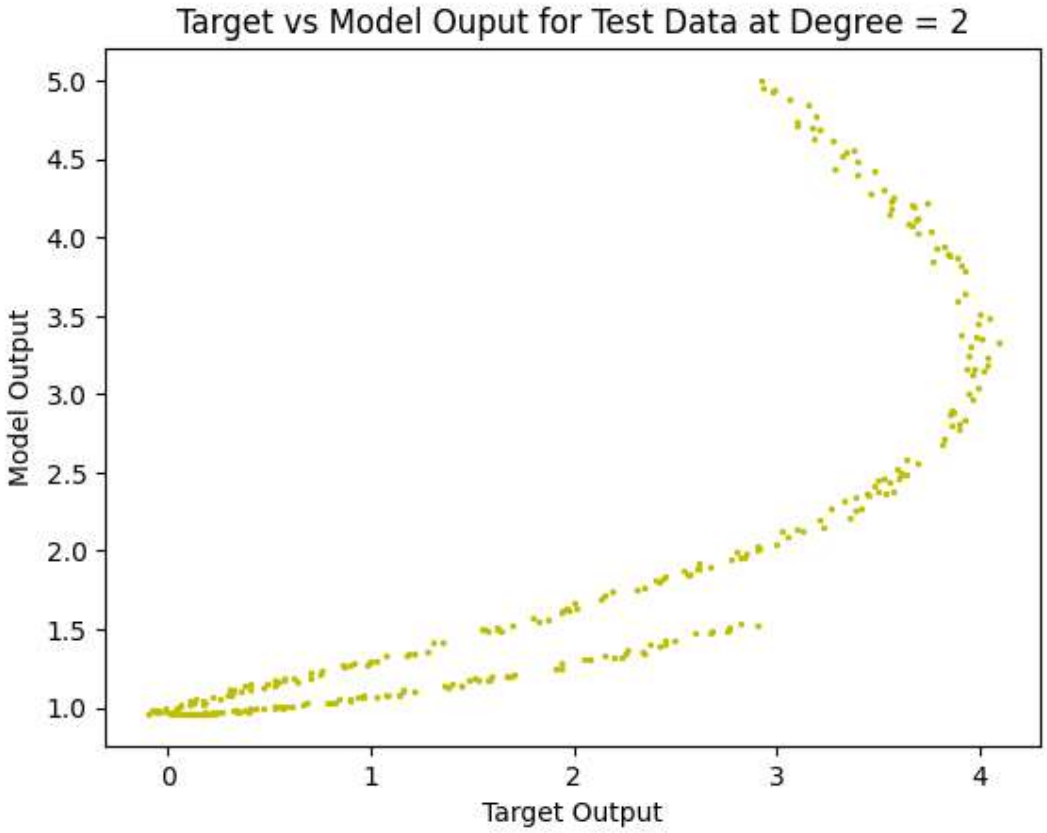
And as we see that is the case for results of degrees we see ahead.

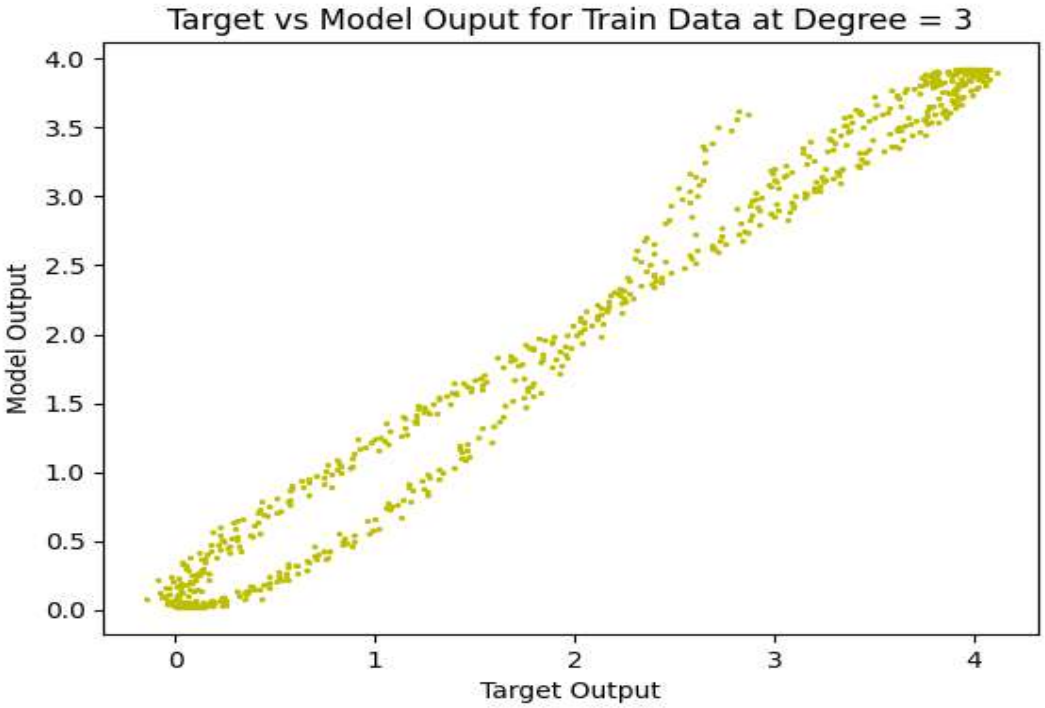
The plots for regularisation (if needed) for a certain number of training samples and for certain degrees have been presented at the end (after all the observations for 700 samples, 100 samples, 50 samples and 10 samples).

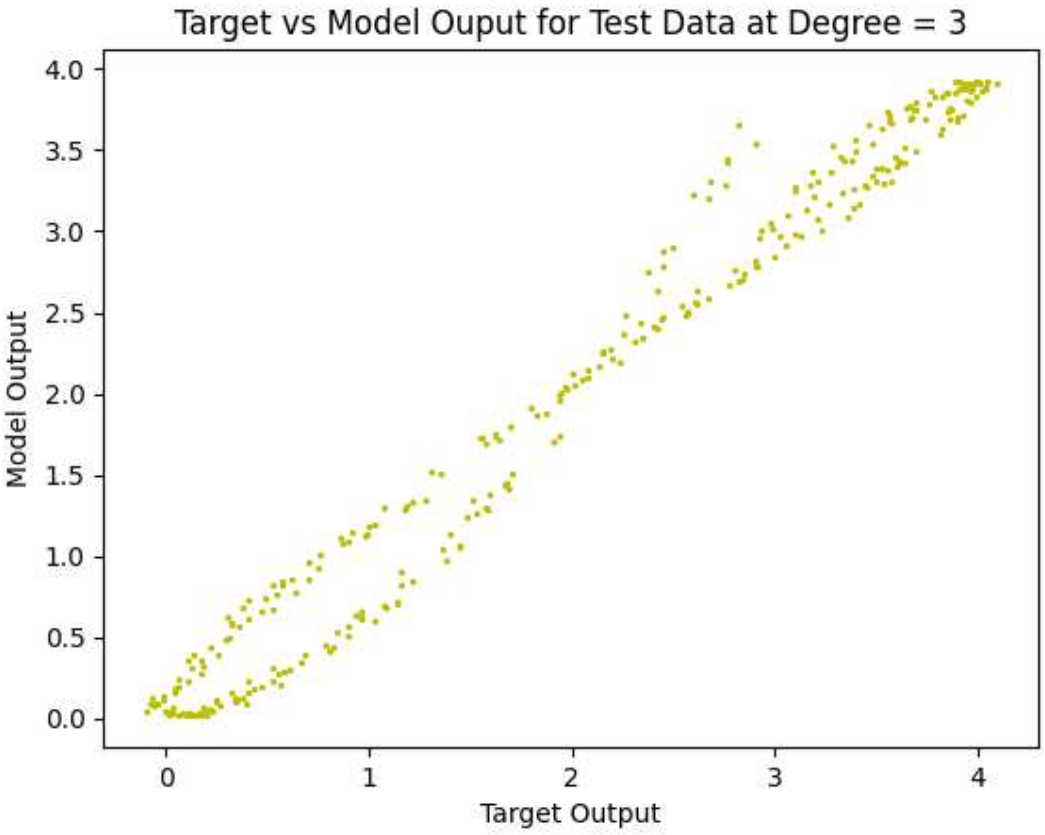
Scatter plot with target output on x-axis and model output on y-axis, for training data, and test data. (for Dataset 1). Give the plots for the best model after cross validation.

Train samples are 700 and test samples are 300.

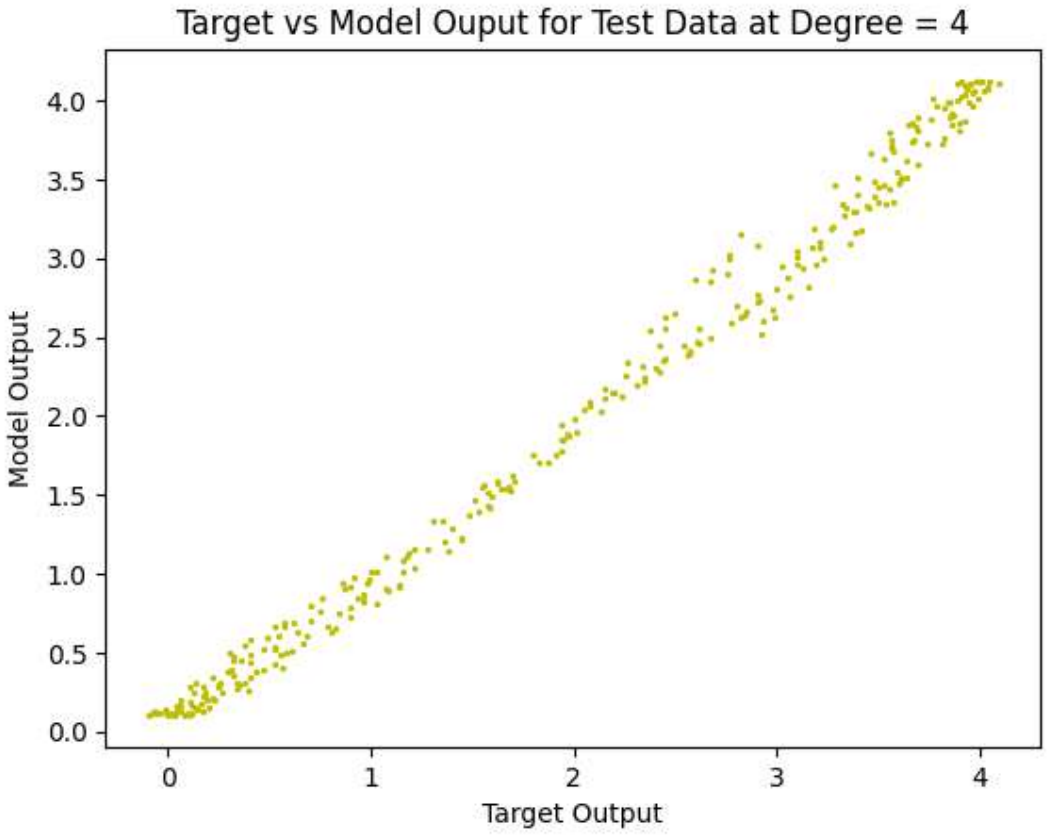


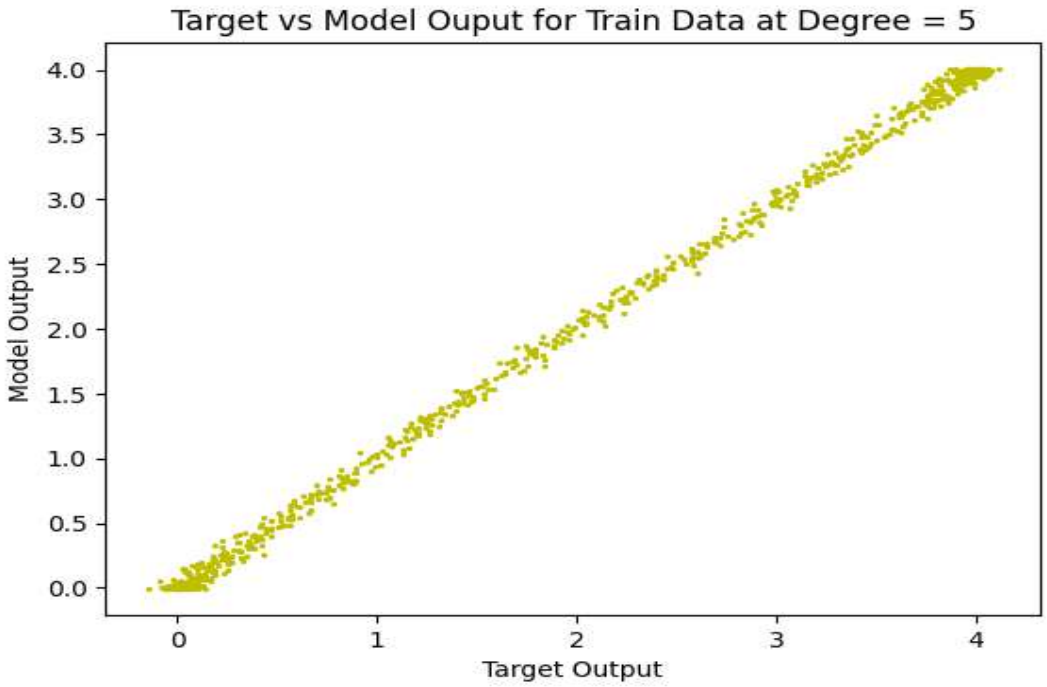


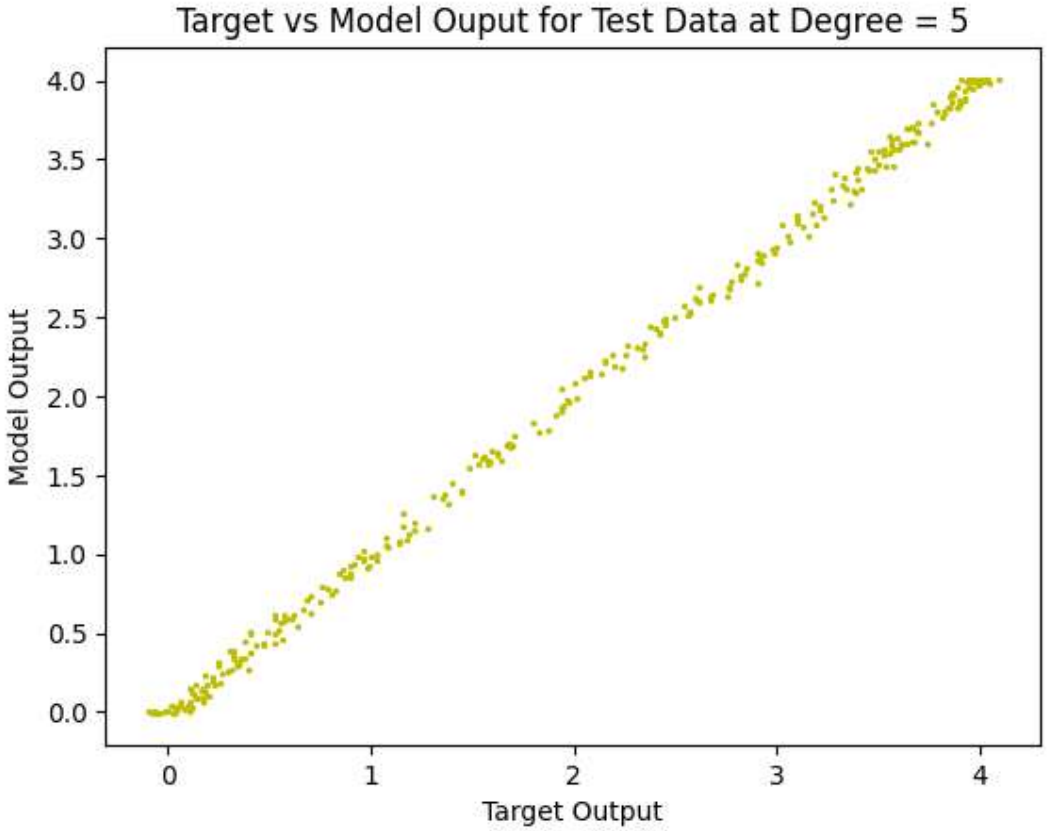

Target vs Model Ouput for Train Data at Degree = 3
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

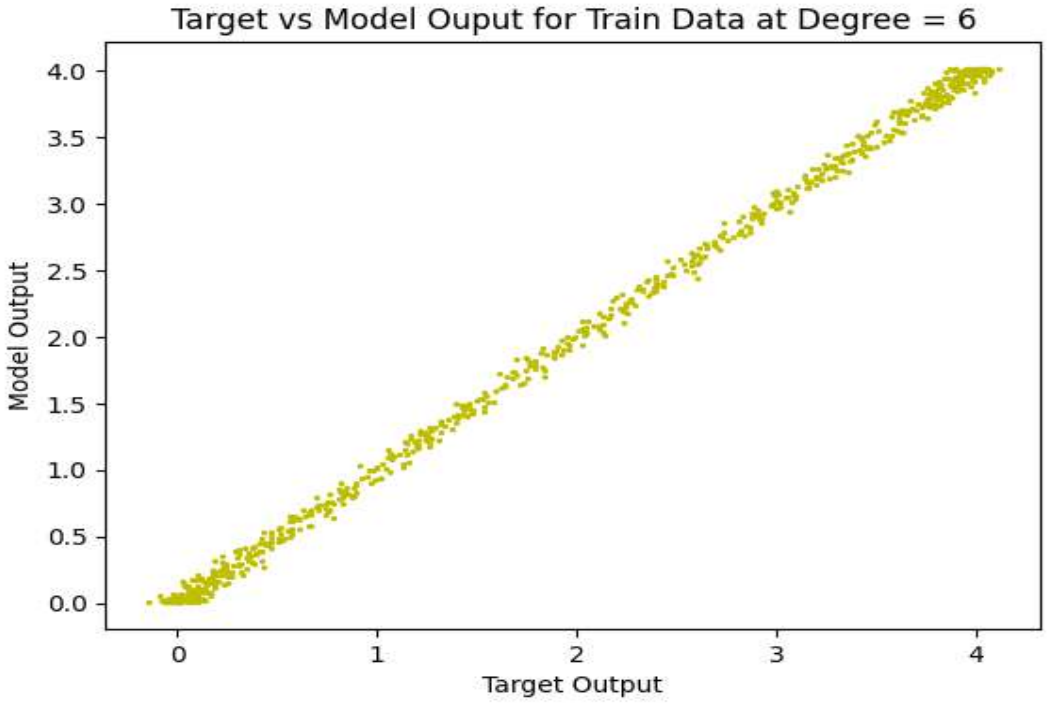

Target vs Model Ouput for Test Data at Degree = 3
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

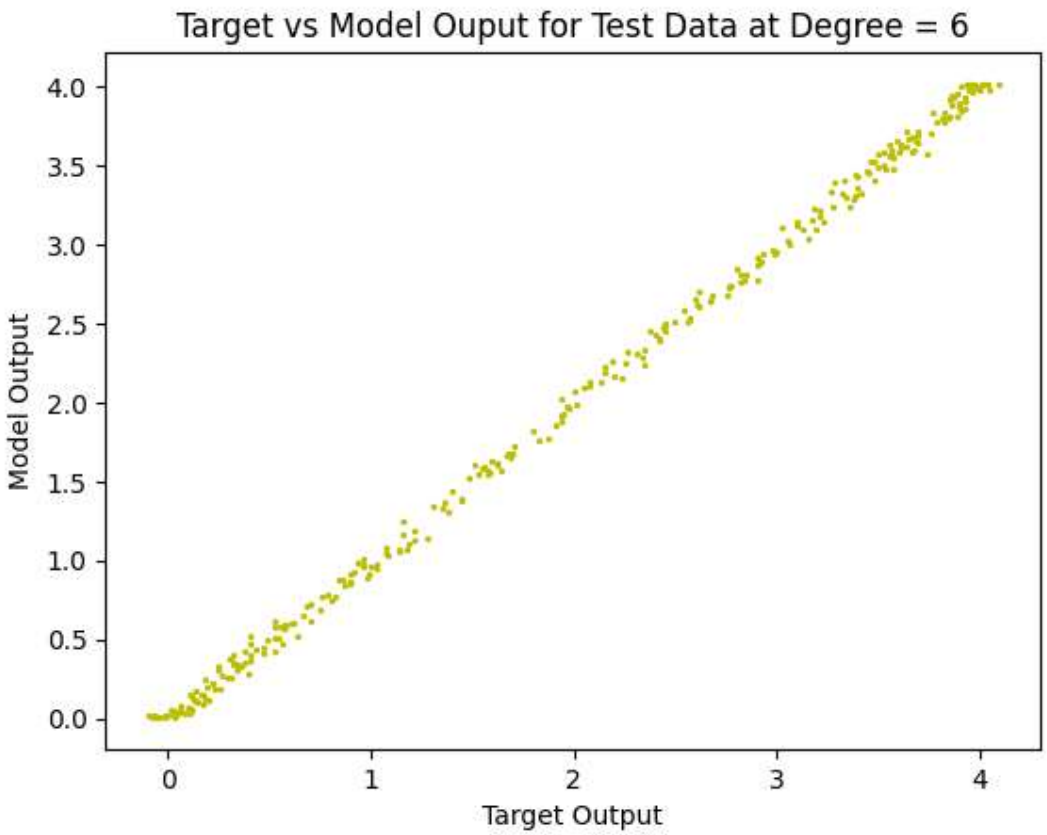

Target vs Model Ouput for Train Data at Degree = 4
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

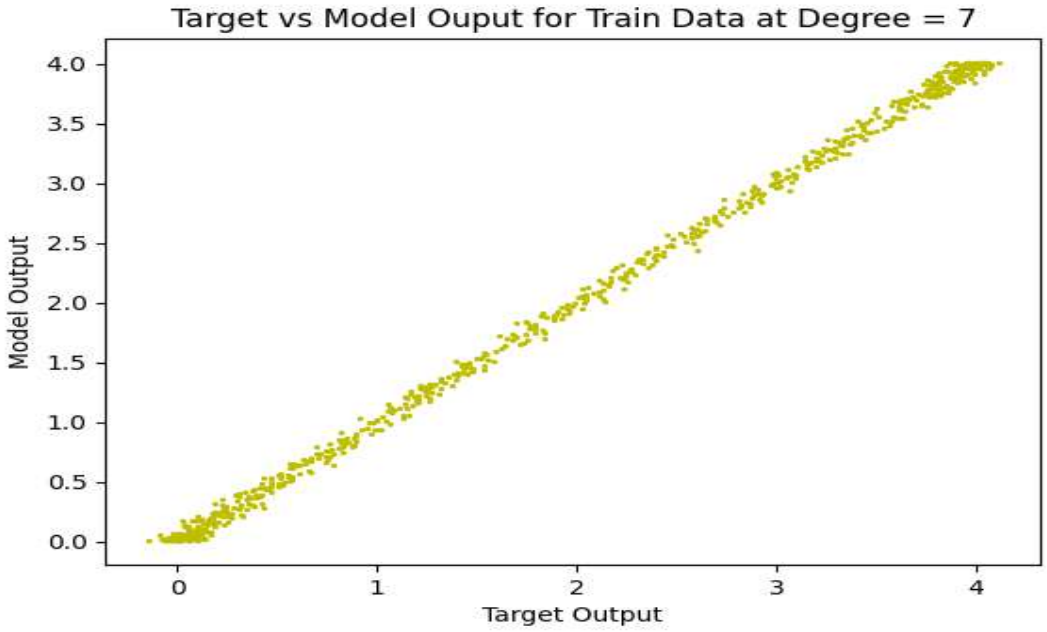

Target vs Model Ouput for Test Data at Degree = 4
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

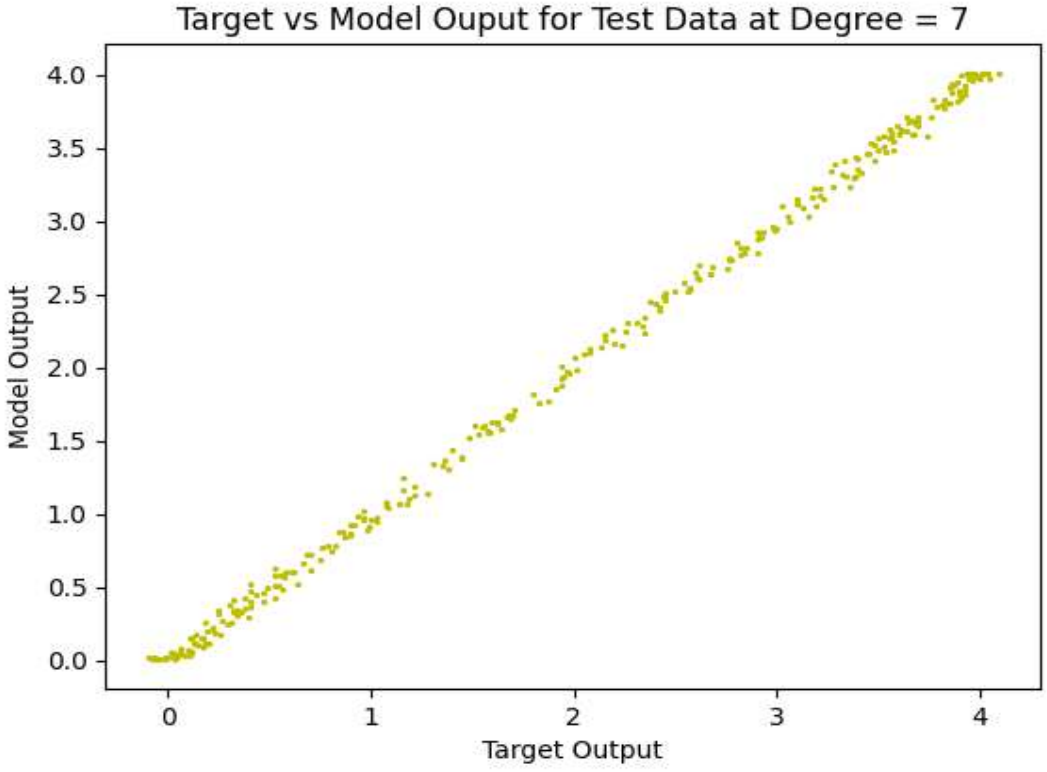

Target vs Model Ouput for Train Data at Degree = 5
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

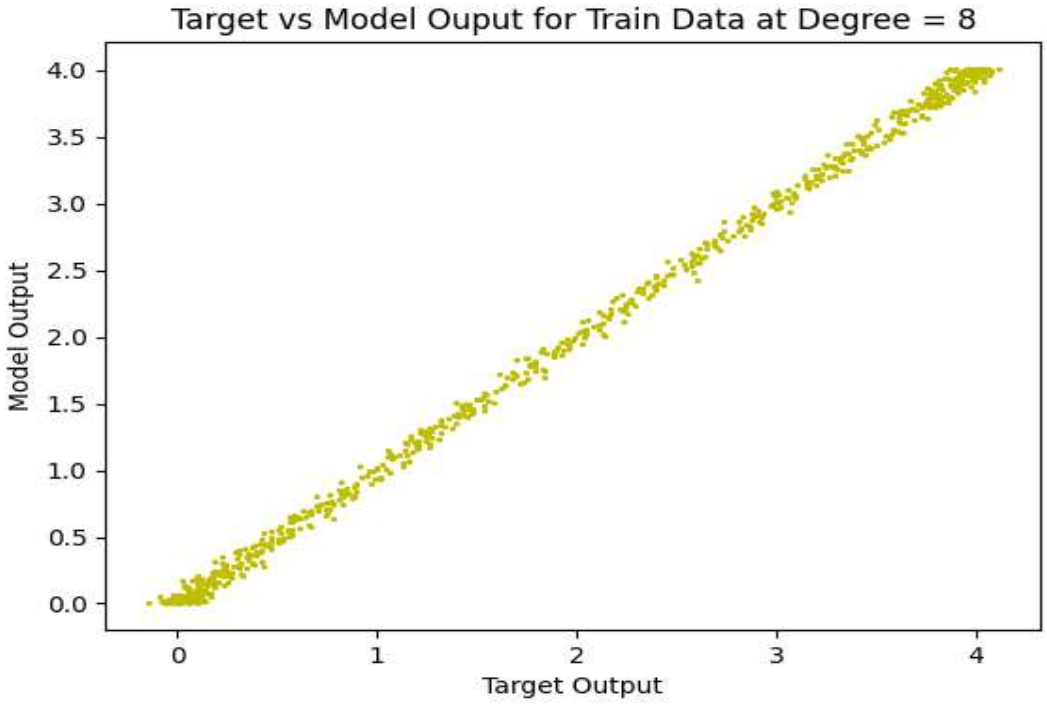

Target vs Model Ouput for Test Data at Degree = 5
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

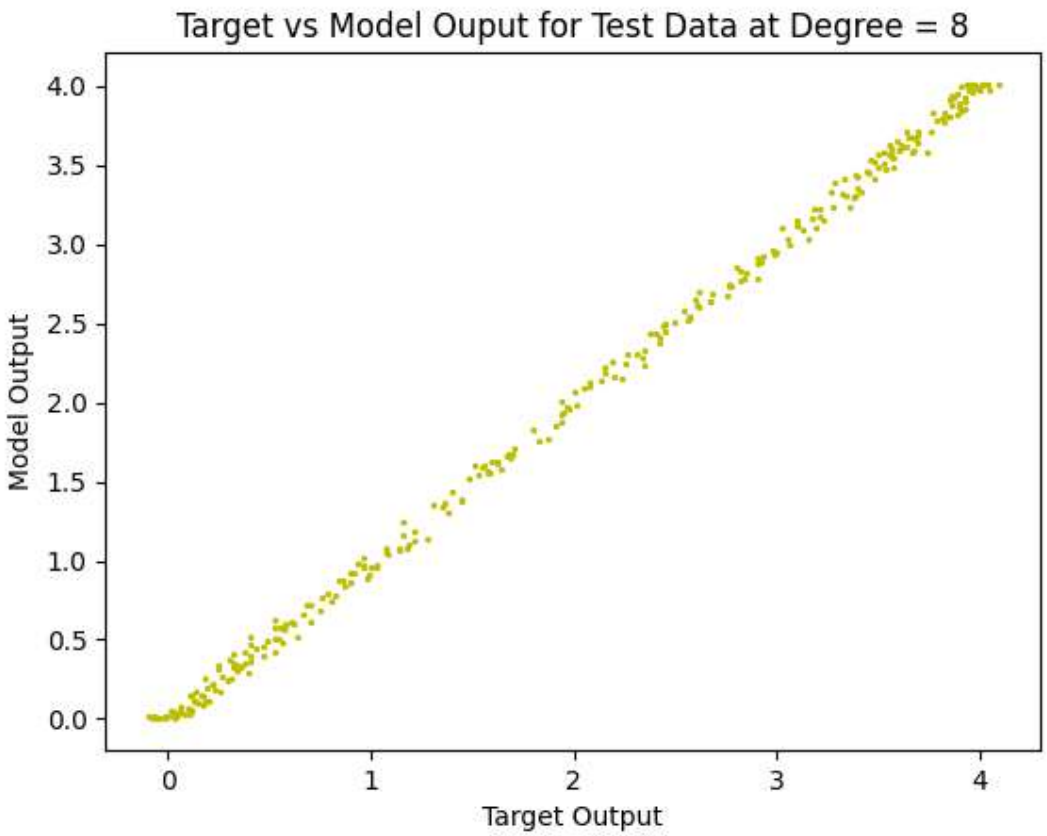

Target vs Model Ouput for Train Data at Degree = 6
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

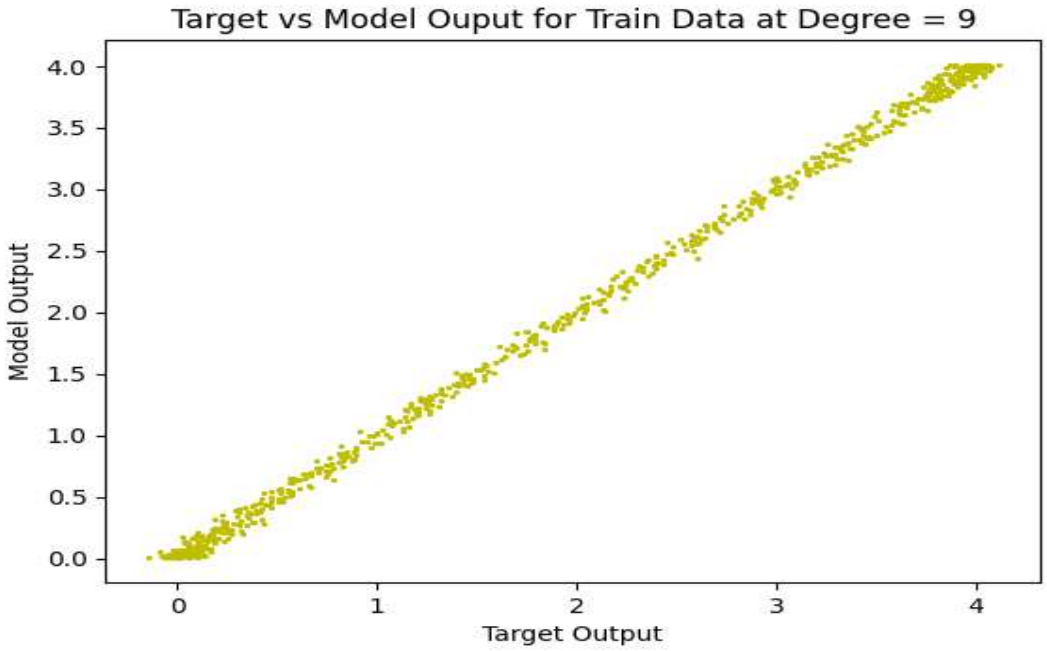

Target vs Model Ouput for Test Data at Degree = 6
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

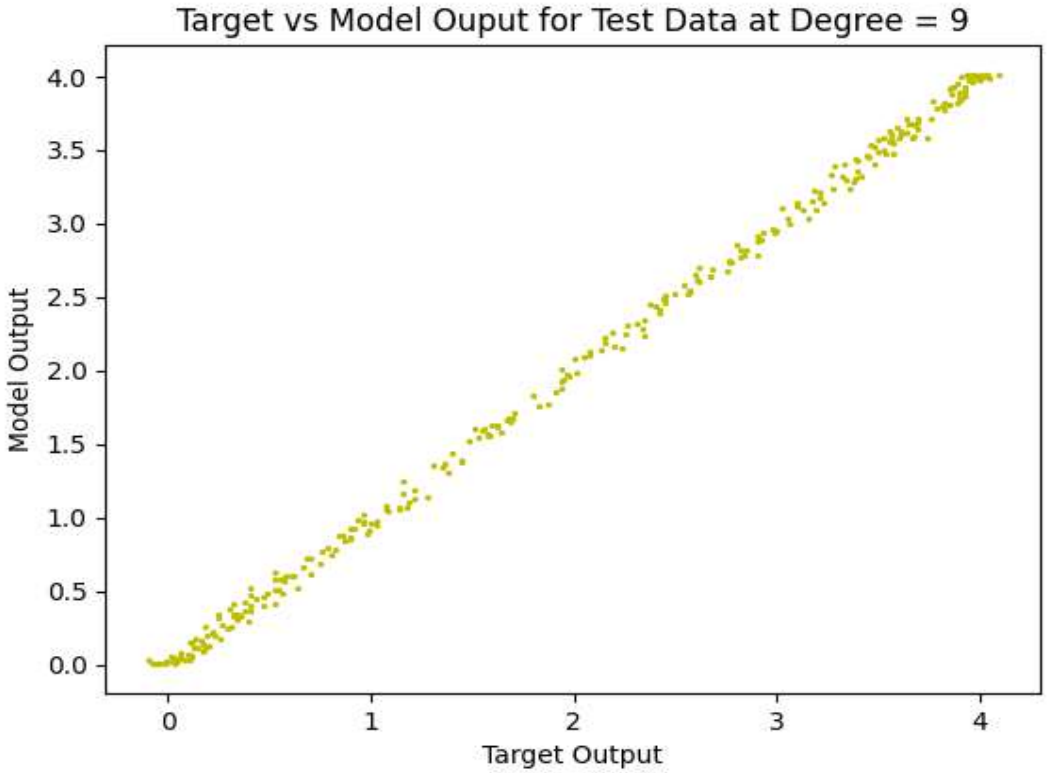

Target vs Model Ouput for Train Data at Degree = 7
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4


Target vs Model Ouput for Test Data at Degree = 7
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4


Target vs Model Ouput for Train Data at Degree = 8
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4


Target vs Model Ouput for Test Data at Degree = 8
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4


Target vs Model Ouput for Train Data at Degree = 9
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

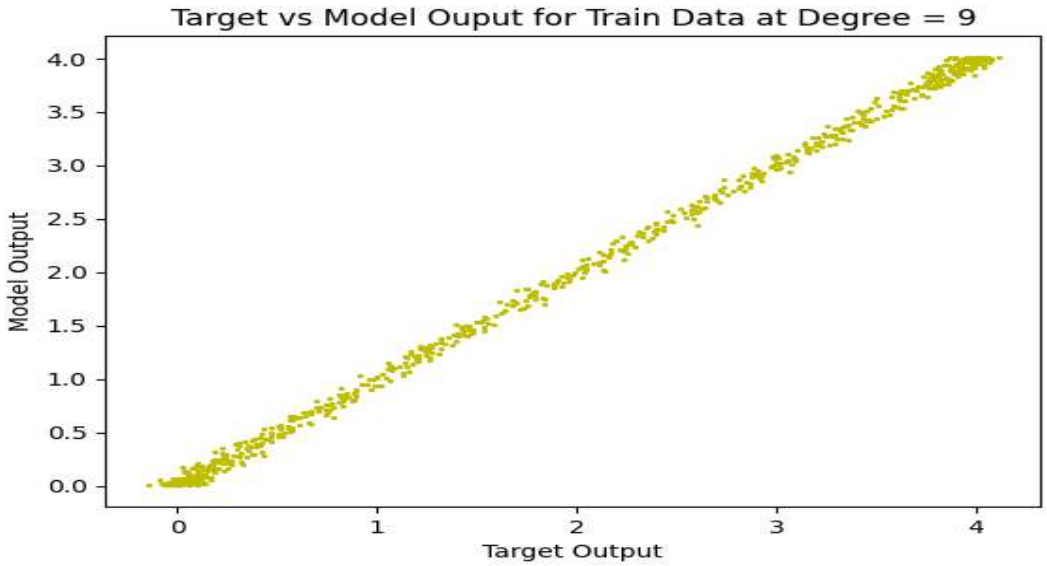

Target vs Model Ouput for Test Data at Degree = 9
Model Output
Target Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

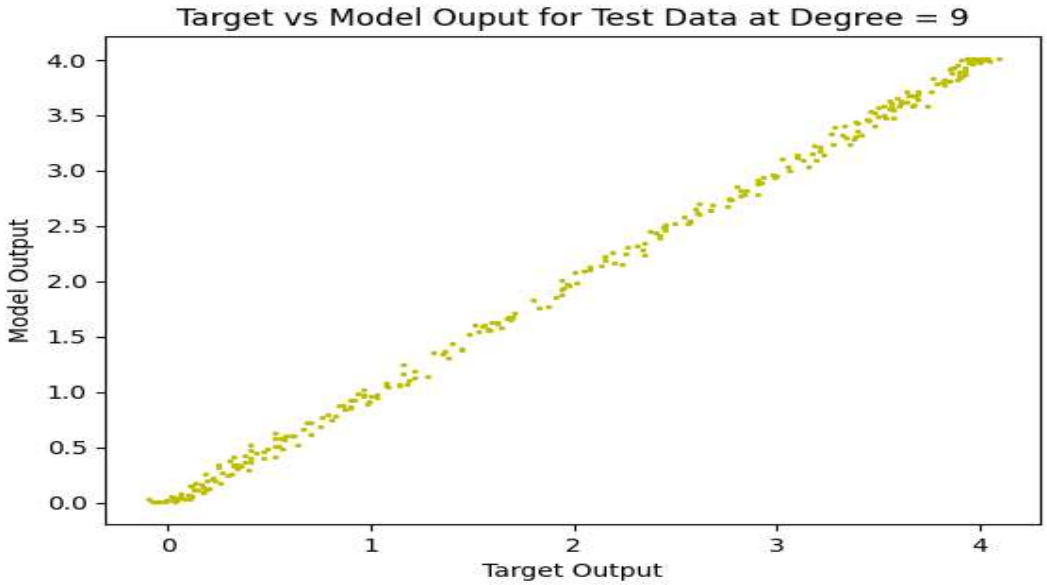
Conclusion by observing scatter plots (Target output Vs Model output) for test (300 samples) and train data (700 samples).

After cross validation we clearly observe a straight line for 9 degree for train and test samples.

Actually, from degree 5 onwards till 9 we almost obtain a straight line for train and test sets for 700 train samples and 300 test samples.

After cross validation we see that for degree 9 the target output resembles predicted output for train and test sets.



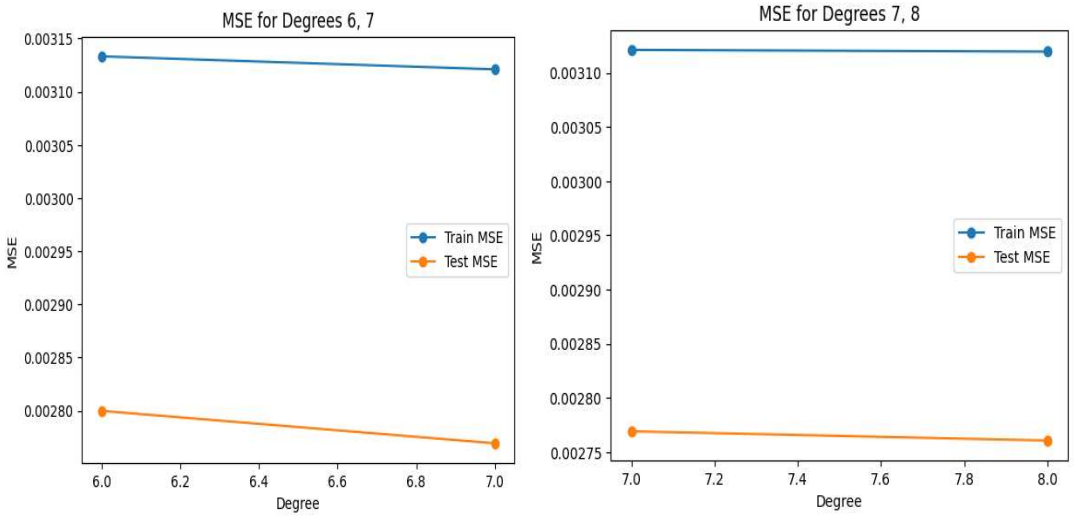


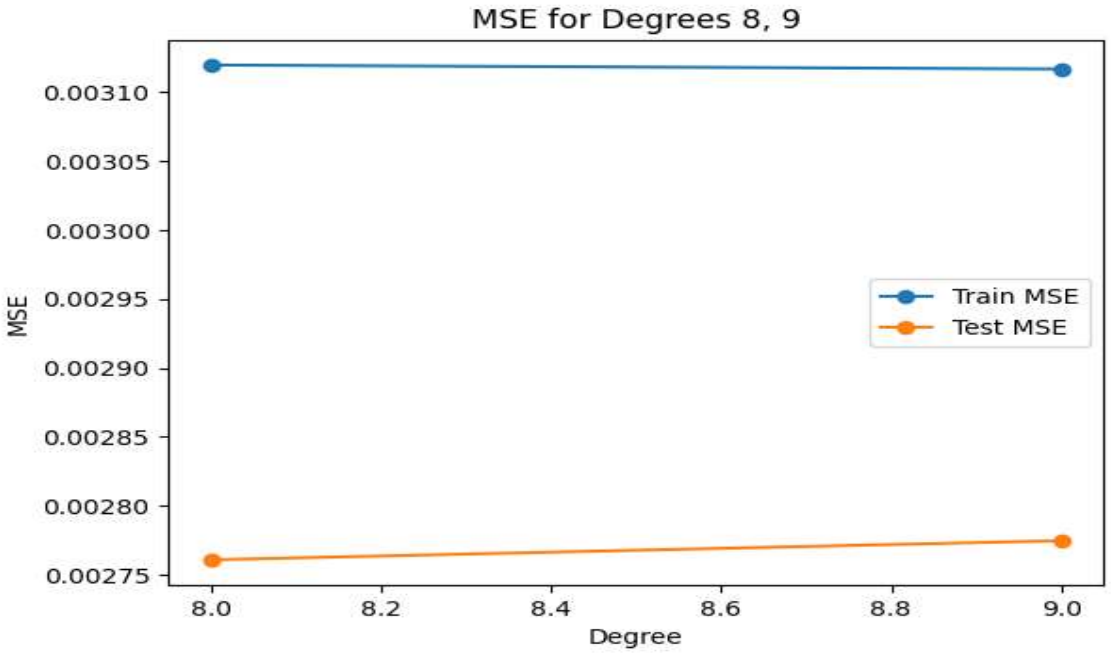
Best scatter plot (resemblance of target and actual output for train and test sets) is for degree 9.

Regularization:

For regularization we take a close look at our MSE plots but we do it piecewise.

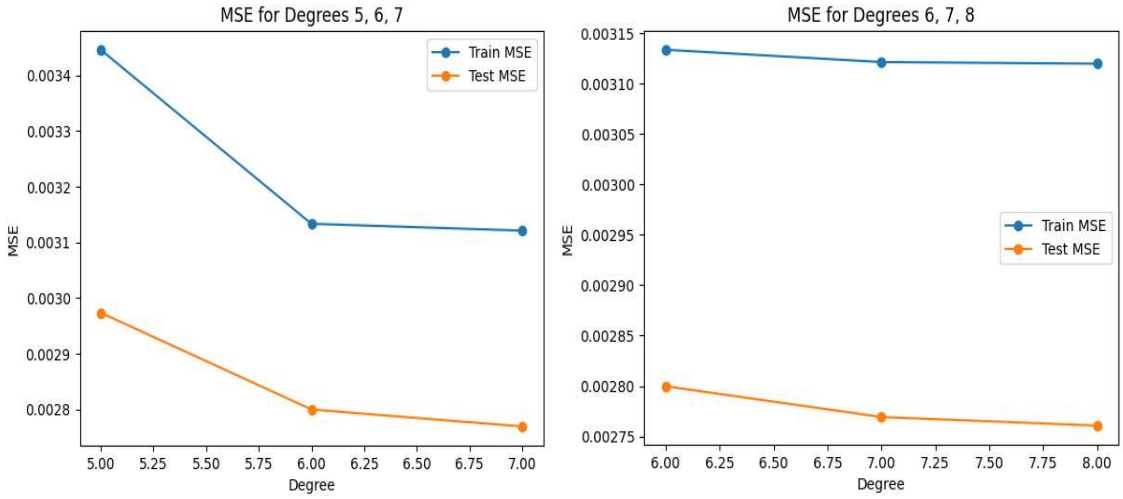
For 700 samples we have few piecewise MSE curves

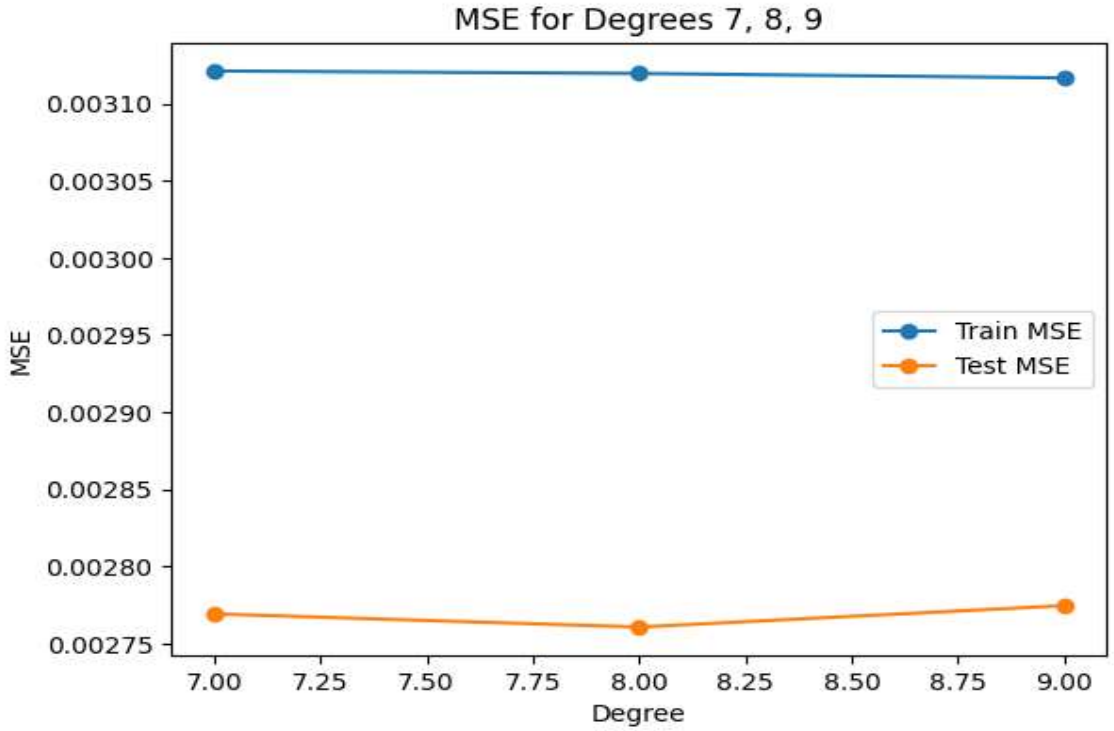




If we just see the graphs of MSE for degrees 6, 7 then 7, 8 and finally 8, 9 we observe an increase in test MSE from 8 to 9 whether it is actually true can be observed as we plot MSE for 3, 4, 5, 6 and 7 degrees together.

MSE curves for 3 degrees combined

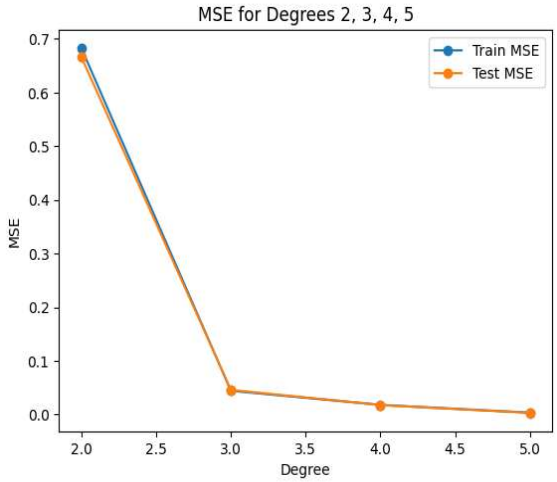


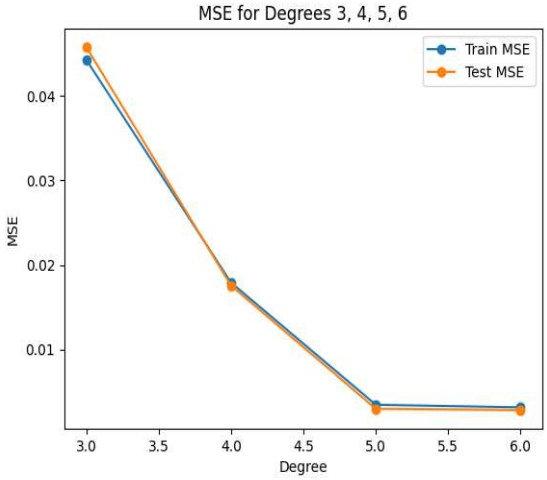


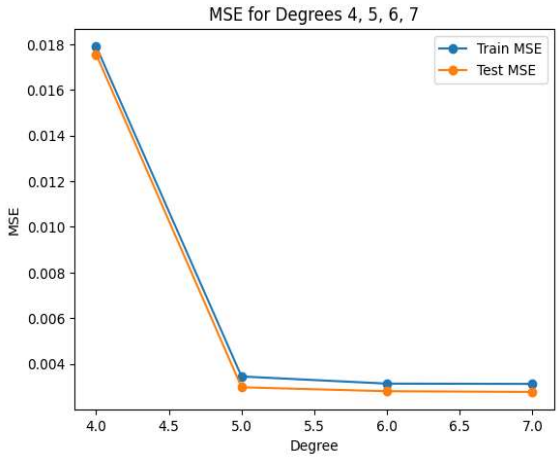
While plotting triplets as well we observe that we have obtained increase in the test MSE from degree 8 to 9.

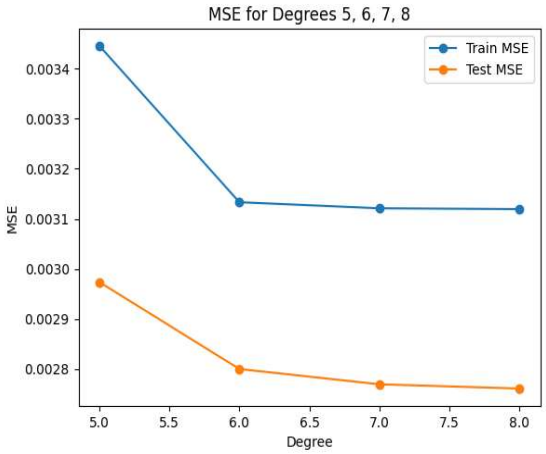
Let us continue observing it further.

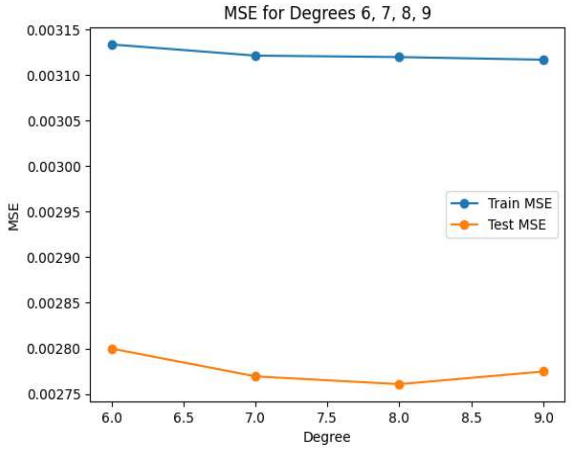
MSE curves for 4 degrees combined. This time we plot all the possible combinations of 4 degrees (in series).







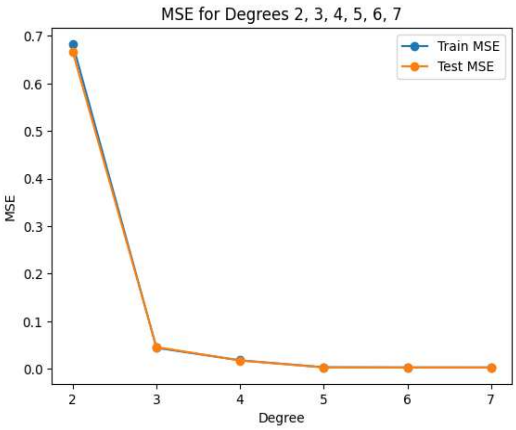


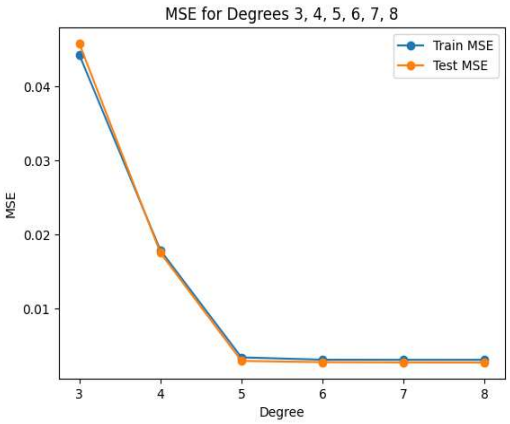


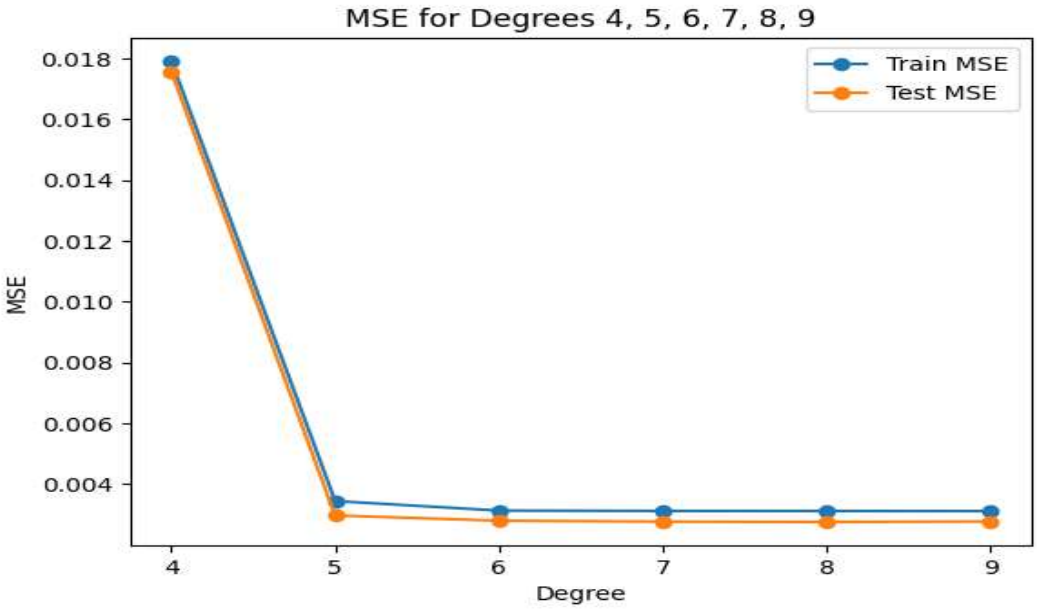
What we observe still is that our test MSE is only increasing from 8 to 9 degree.

Let us keep on observing.

MSE curves for 6 degrees combined. This time we plot all the possible combinations of 6 degrees (in series).



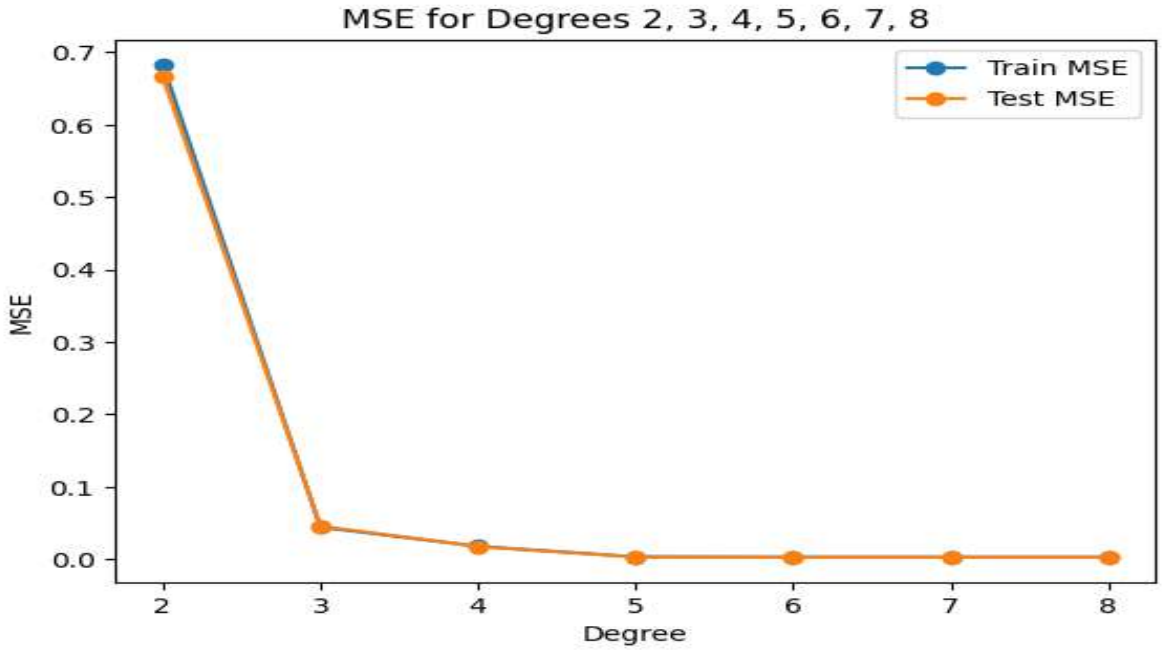


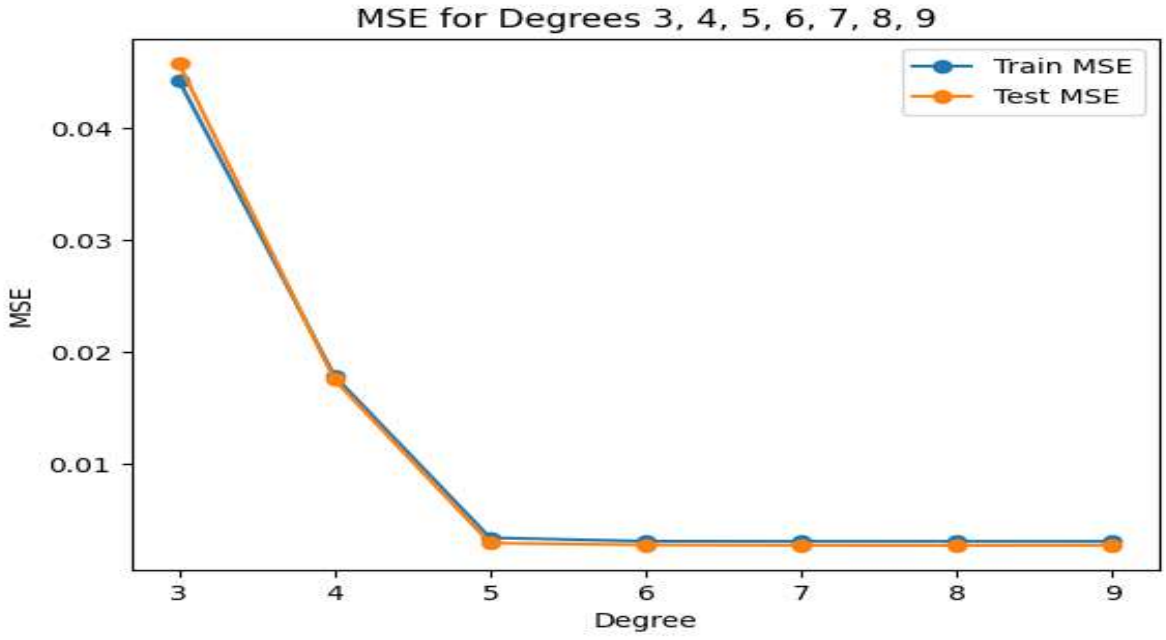


The reason why we gave MSE plots for combination of 2 and 3 degrees is because though test MSE has increased from 8 to 9 but here we can't observe it when we are plotting the MSE curves for a combination of 5, 6, 7, 8, or 9 serial degrees combined.

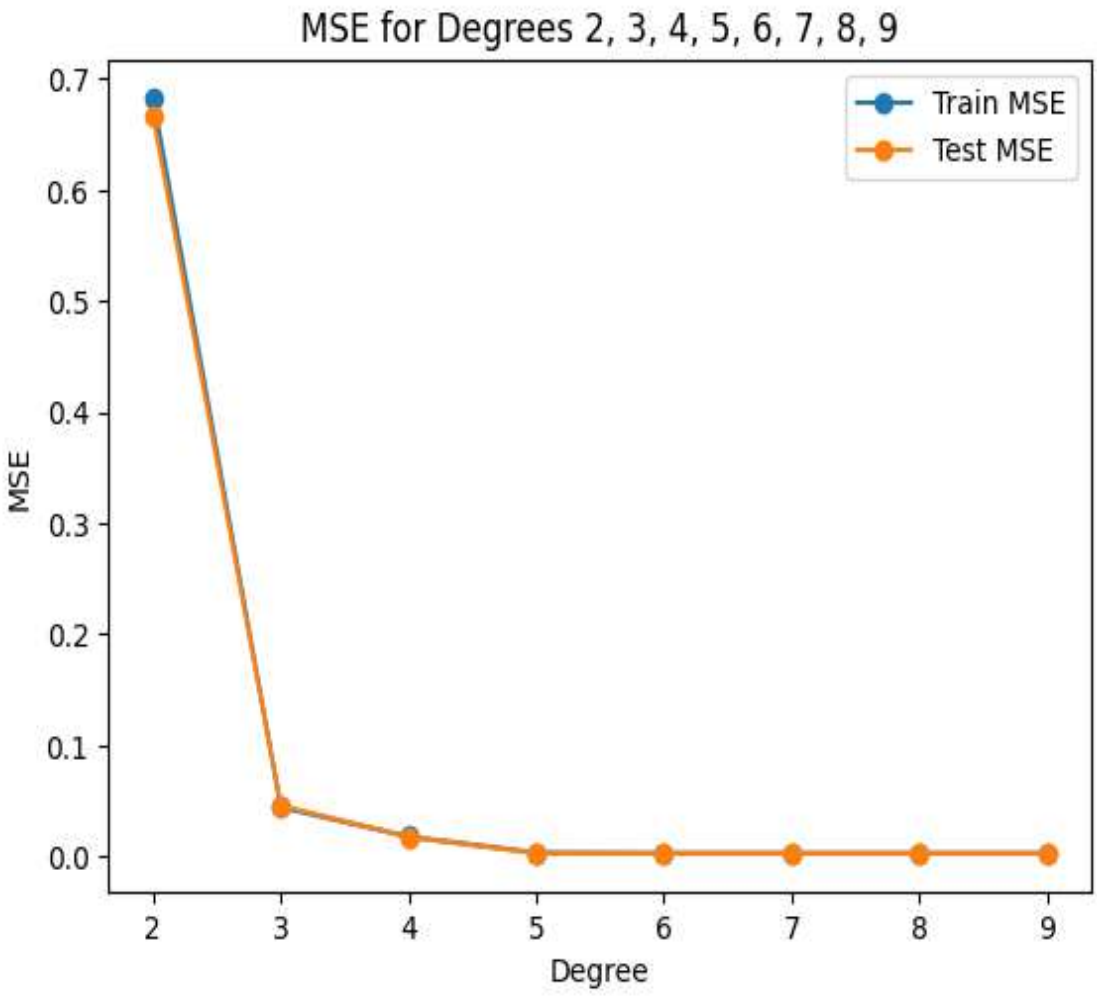
So, we can now claim that out test MSE has increased from degree 8 to 9.

MSE curves for 7 degrees combined. This time we plot all the possible combinations of 7 degrees (in series).



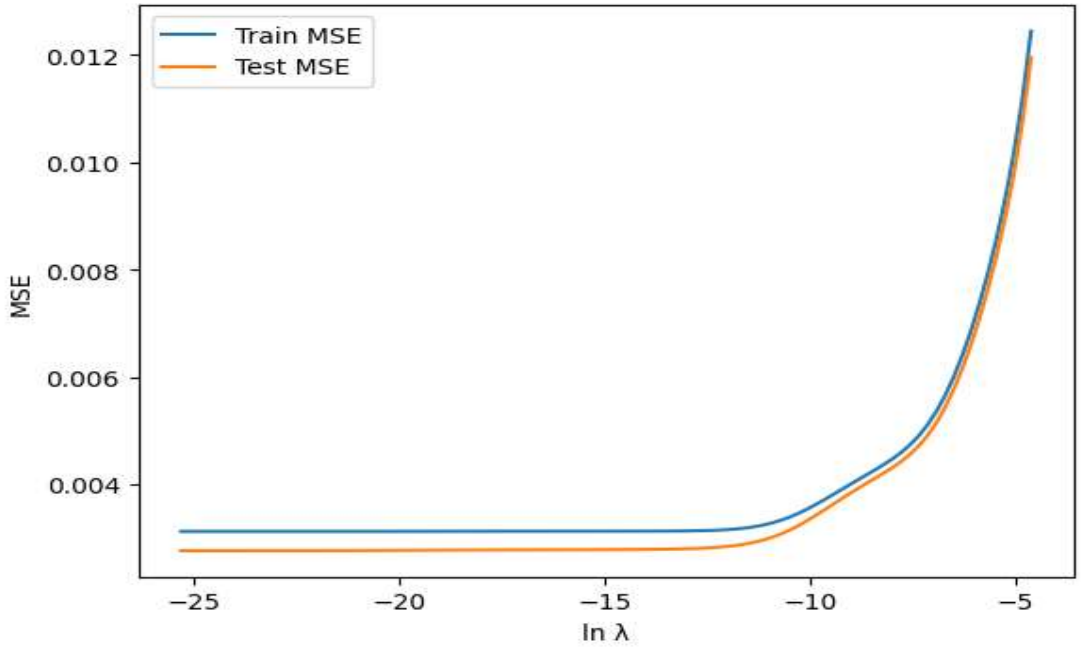


Cumulative MSE plot

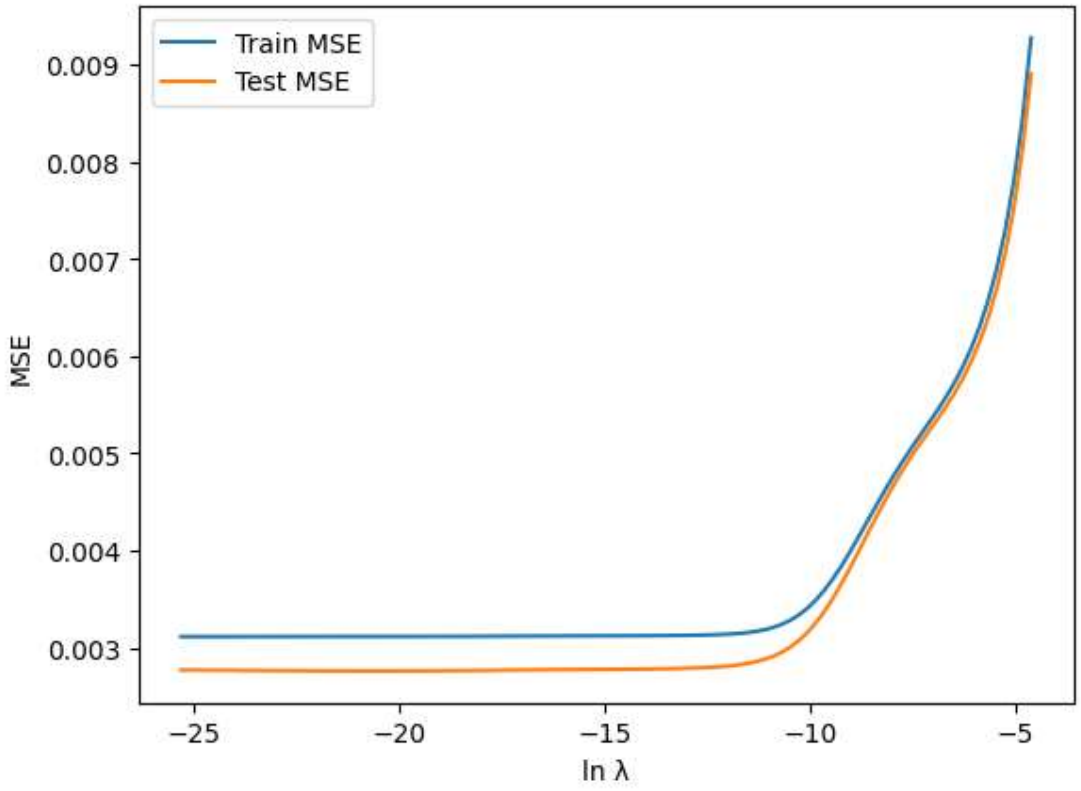


Now we perform regularization for degree 9 for the case of 700 train samples and 300 test sample.

Regularization for degree 9 when number of train samples is 700 and test samples is 300

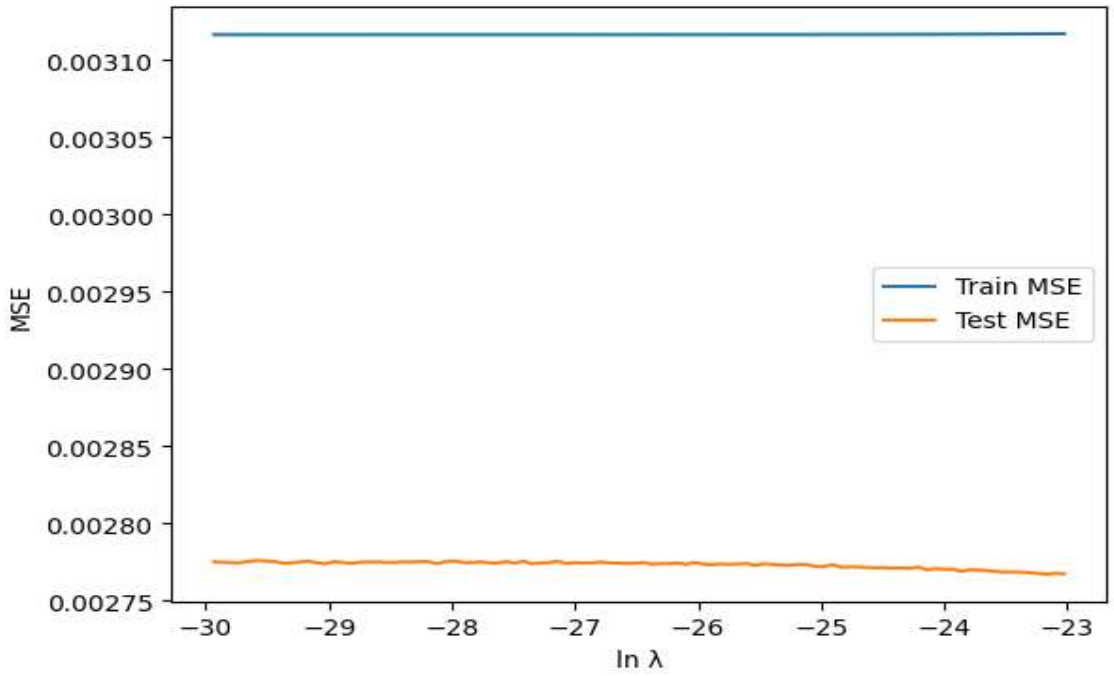


Now we select values of that lambda where test MSE is the lowest.



We run a code to find that value of lambda where our test MSE is minimum.

Here is that particular graph which has helped us find the global minima of test MSE for graph of ln lambda VS train and test MSE.




```
print(min(mse_test))              0.002767166296820939
print(min(mse_train))             0.003116531717700654
print(np.argmin(mse_train))       12
print(np.argmin(mse_test))        97
print(alpha_values[12])           2.310129700083158e-13
print(alpha_values[97])           8.697490026177835e-11
```

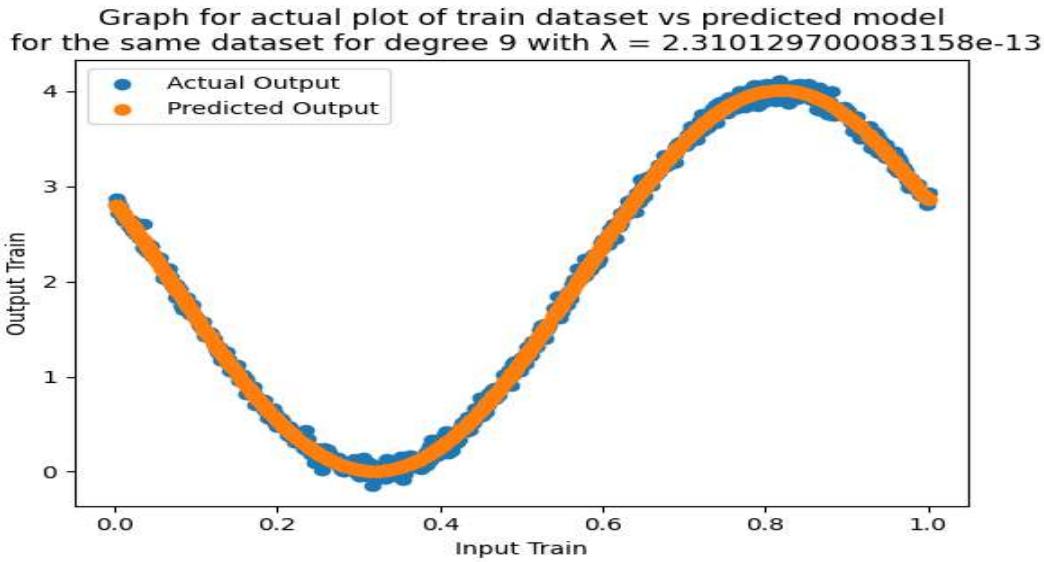
We observe that our minimum mse_test is `0.002767166296820939.`
Earlier we observed that the test mse for degree 9 was
`MSE for degree 9 test data: 0.00277467618376161.`

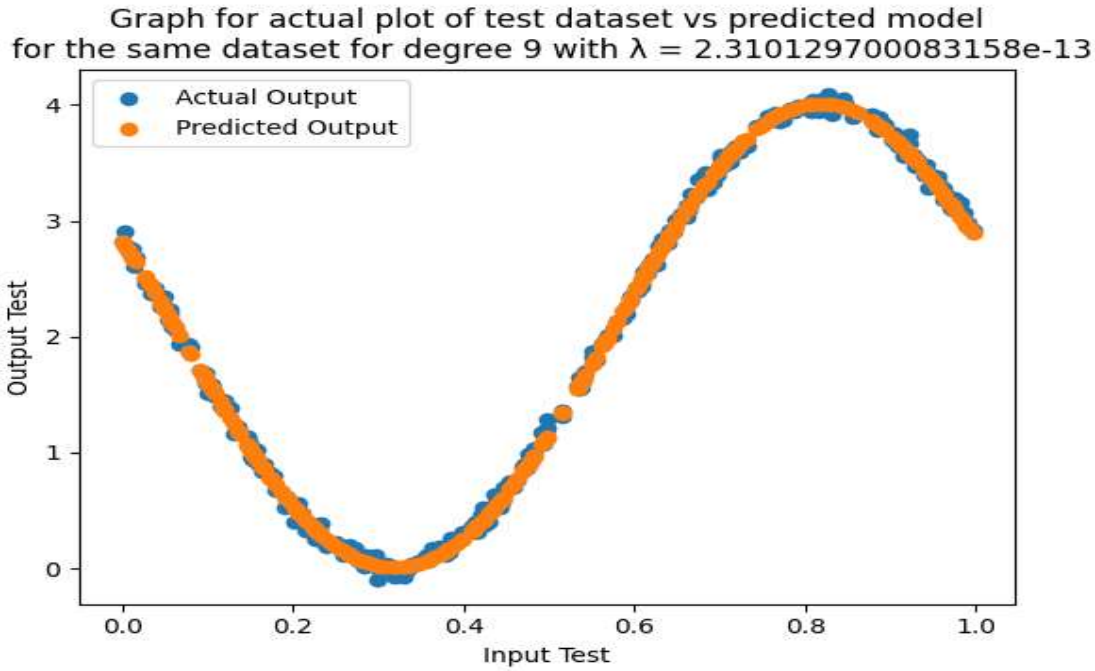
We observe that our minimum mse_train is 0.003116531717700654.
Earlier we observed that the train mse for degree 9 was
`MSE for degree 9 train data:` 0.003116534707622286.

Essentially both train and test MSE for a certain value of lambda while regularising is lower than that of train and test MSE values without regularisation.

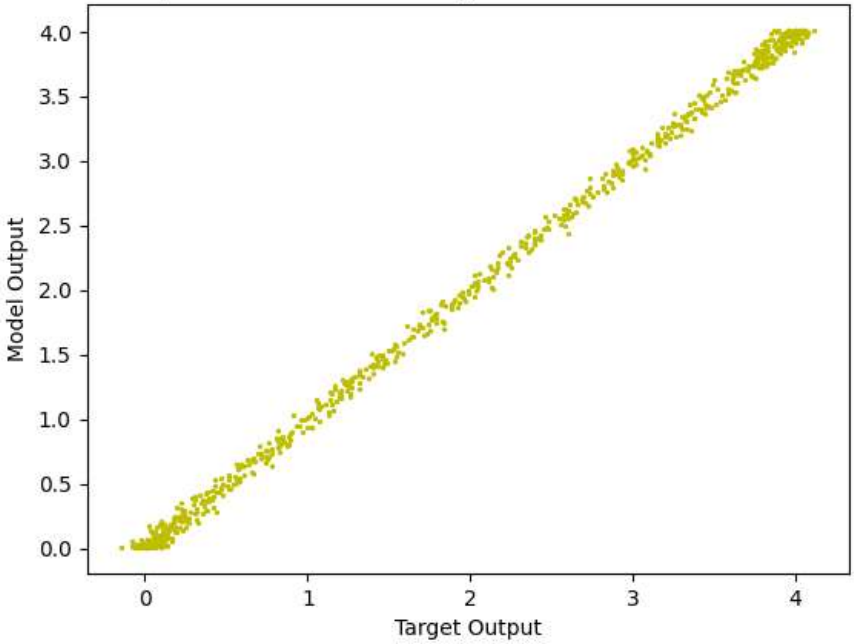
2 values of lambda for which we have this train and test MSE as the minimum are:
2.310129700083158e-13
8.697490026177835e-11

After regularization we obtain the following plot for **lambda=2.310129700083158e-13**

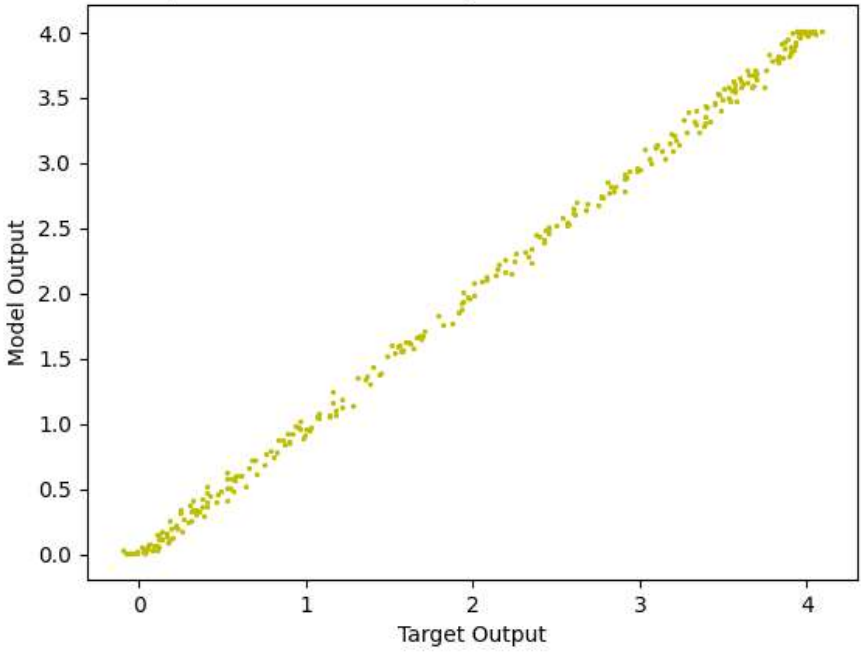


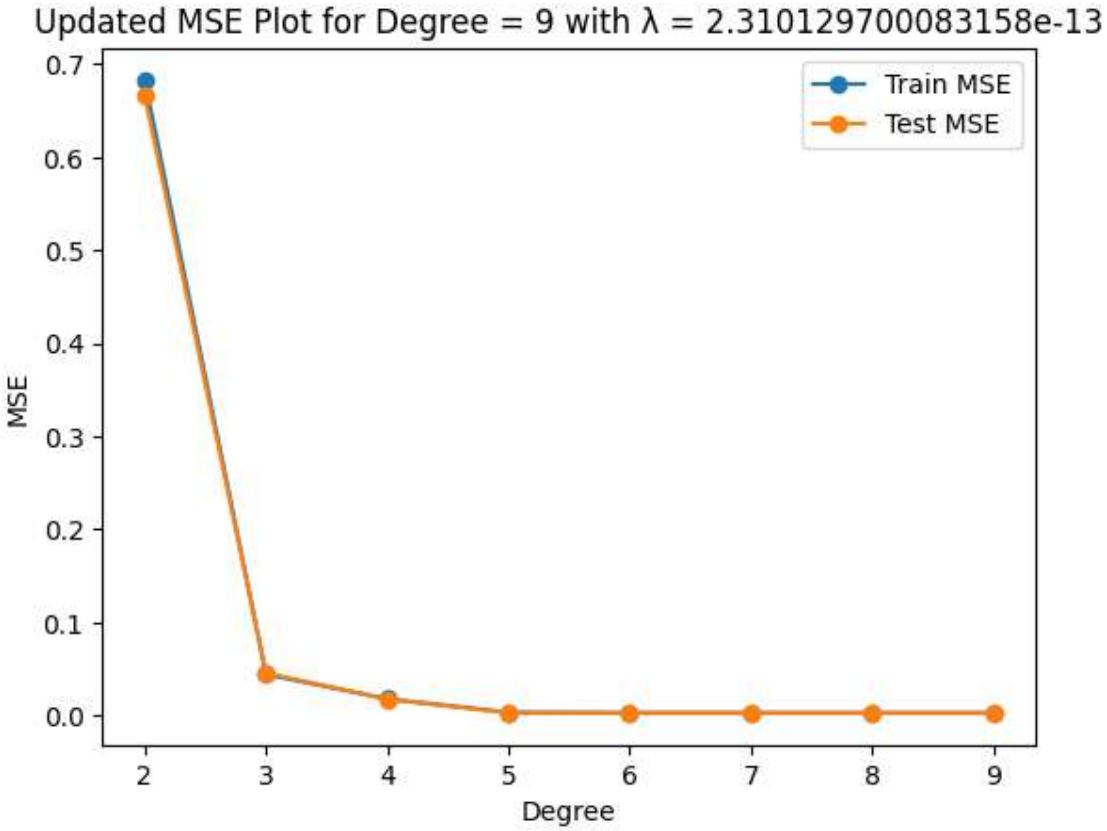


Target vs Model Ouput for Train Data at Degree = 9 with λ = 2.310129700083158e-13



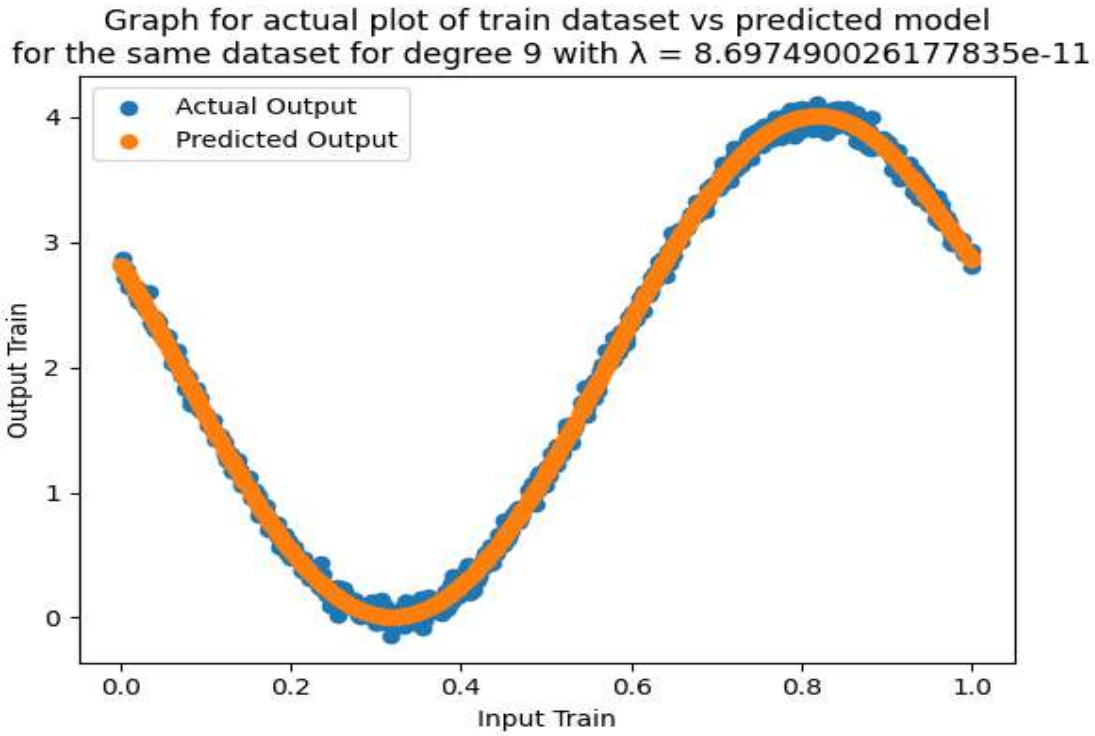
Target vs Model Ouput for Test Data at Degree = 9 with λ = 2.310129700083158e-13

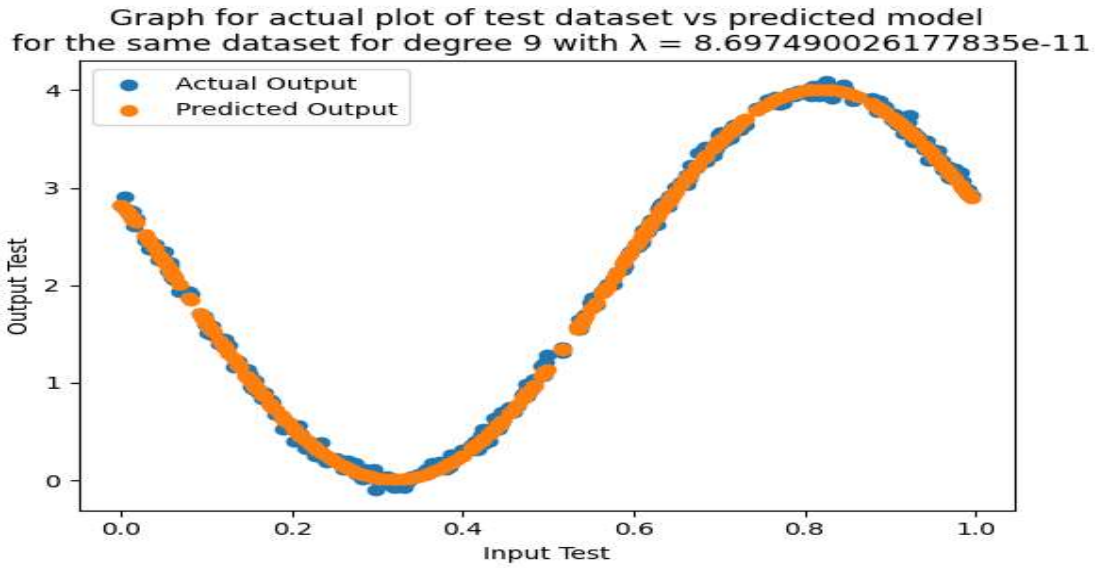




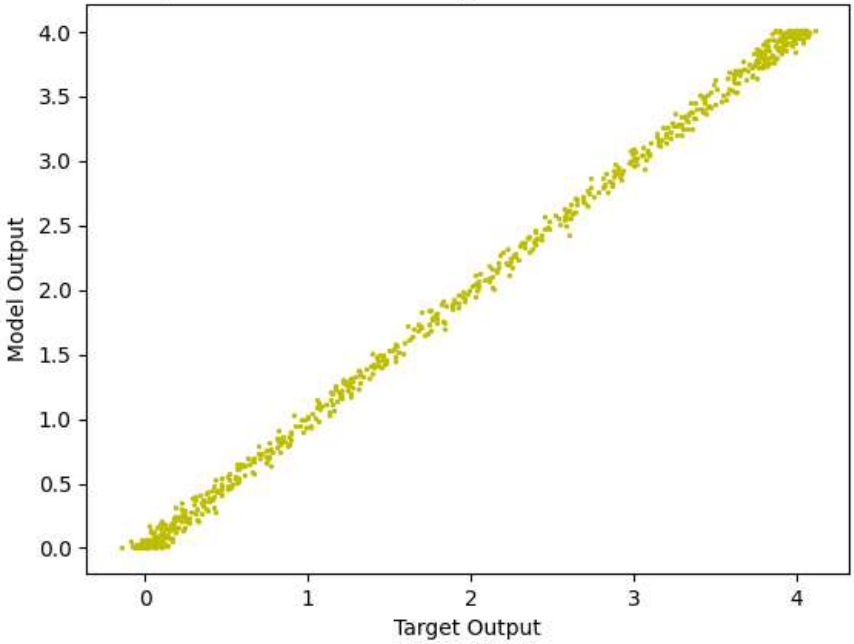
If we zoom it and only see the graph for 7 and 8, 8 and 9 degrees we will observe that the test MSE does not increase from 8 to 9 after regularisation.

After regularization we obtain the following plot for **lambda= 8.697490026177835e-11**

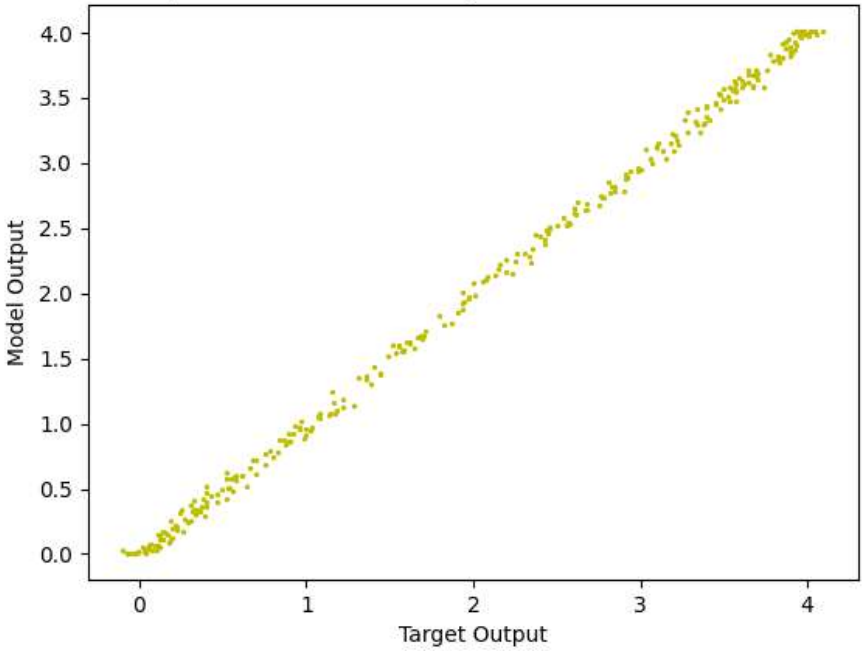


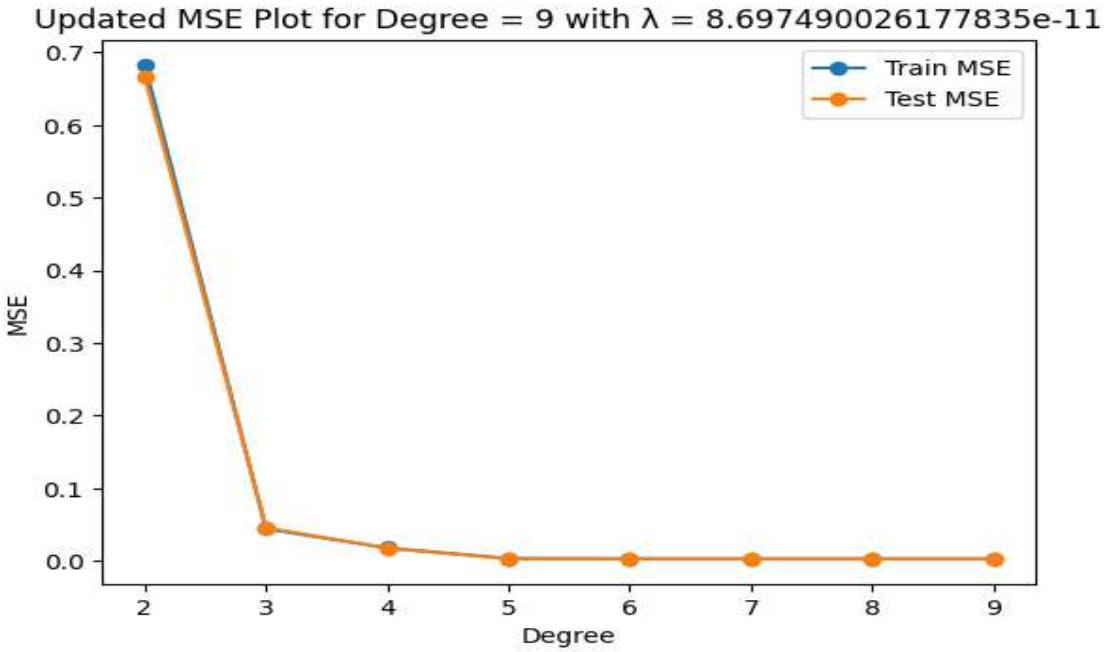


Target vs Model Ouput for Train Data at Degree = 9 with λ = 8.697490026177835e-11



Target vs Model Ouput for Test Data at Degree = 9 with λ = 8.697490026177835e-11





Observations:

For lambda=2.310129700083158e-13

```
w0 value of degree 9 2.8198617212403834
w1 value of degree 9 -10.608896447352762
w2 value of degree 9 -28.961871381614174
w3 value of degree 9 176.34087027066744
w4 value of degree 9 -418.40309636043094
w5 value of degree 9 1195.1988160593173
w6 value of degree 9 -2451.6100700028474
w7 value of degree 9 2723.7413664900523
w8 value of degree 9 -1542.2650218377385
w9 value of degree 9 356.60496412840075
```

For lambda=8.697490026177835e-11

```
w0 value of degree 9 2.823662609652052
w1 value of degree 9 -10.933300574161116
w2 value of degree 9 -22.045838744180287
w3 value of degree 9 113.34902451955986
w4 value of degree 9 -117.55654560710536
w5 value of degree 9 367.5536764032295
w6 value of degree 9 -1093.2199308352428
w7 value of degree 9 1410.9314132903528
w8 value of degree 9 -853.1507677224436
w9 value of degree 9 205.1031826494609
```

For lambda=2.310129700083158e-13

```
MSE for degree 9 training data: 0.003116531717700654
MSE Percent for degree 9 training data: 0.15531812907785353
MSE for degree 9 test data: 0.0027744785677756715
MSE Percent for degree 9 test data: 0.13871528976737993
```

For lambda=8.697490026177835e-11

```
MSE for degree 9 training data: 0.0031170701162577734
MSE Percent for degree 9 training data: 0.15534496116690785
MSE for degree 9 test data: 0.002767166296820939
MSE Percent for degree 9 test data: 0.13834969898714314
```

Before Regularization

Values of the coefficients and train and test MSE and MSE percent values for 700 train samples and 300 test samples.

w0 value of degree 9:	2.82005592193318
w1 value of degree 9:	-10.608127367657474
w2 value of degree 9:	-28.999180385665795
w3 value of degree 9:	176.6868859096976
w4 value of degree 9:	-420.0512625708943
w5 value of degree 9:	1199.7296789396496
w6 value of degree 9:	-2459.0428296942555
w7 value of degree 9:	2730.9199754823057
w8 value of degree 9:	-1546.030626058564
w9 value of degree 9:	357.43236402791626
MSE for degree 9 training data:	0.003116534707622286
MSE Percent for degree 9 training data:	0.15531827808613455
MSE for degree 9 test data:	0.00277467618376161
MSE Percent for degree 9 test data:	0.13872516995137935

After regularization

For lambda=2.310129700083158e-13

```
MSE for degree 9 training data: 0.003116531717700654
MSE Percent for degree 9 training data: 0.15531812907785353
MSE for degree 9 test data: 0.0027744785677756715
MSE Percent for degree 9 test data: 0.13871528976737993
```

For lambda=8.697490026177835e-11

```
MSE for degree 9 training data: 0.0031170701162577734
MSE Percent for degree 9 training data: 0.15534496116690785
MSE for degree 9 test data: 0.00276716629682093 9
MSE Percent for degree 9 test data: 0.13834969898714314
```

Lowest values of MSE train and test combined is for lambda=2.310129700083158e-13 which is as follows:

```
MSE for degree 9 training data: 0.003116531717700654
MSE Percent for degree 9 training data: 0.15531812907785353
MSE for degree 9 test data: 0.0027744785677756715
MSE Percent for degree 9 test data: 0.13871528976737993
```

Note: When we consider 100 train samples considered for 100 train samples case, those 100 train samples must in turn have 50 train samples considered for the case of 50 train samples case which in turn must have the 10 train samples considered for the 10 train samples case.

Observations for 100 samples:

Plot for train data which is 100 train sample case.

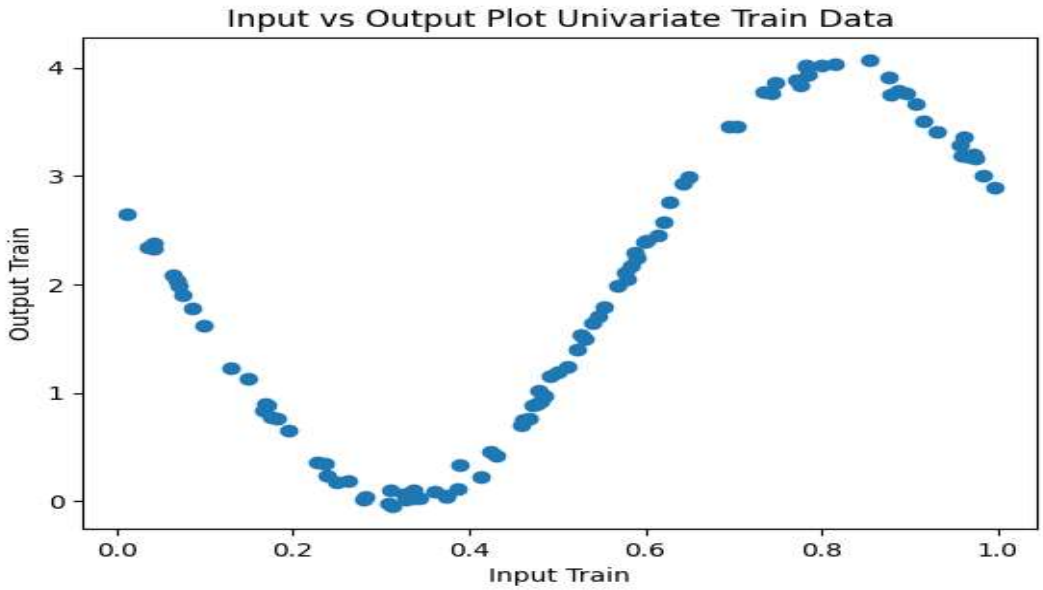


Fig32

- The values of x should roughly lie between 0 and 1 (they being included).
- The y values should roughly lie between -0.14576 and 4.1118 The number of train data points is 700 which is the case where we consider the whole training data (whole 70% of 1001).
- The number of train samples is 100.

Plot of the test data (300 samples) for 100 train samples case.

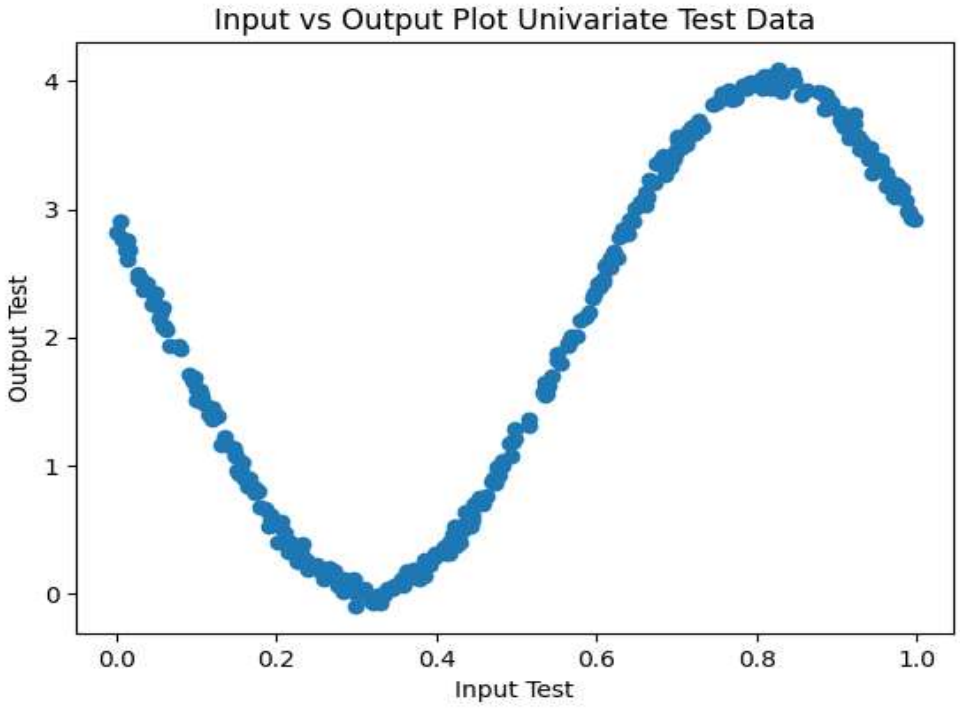
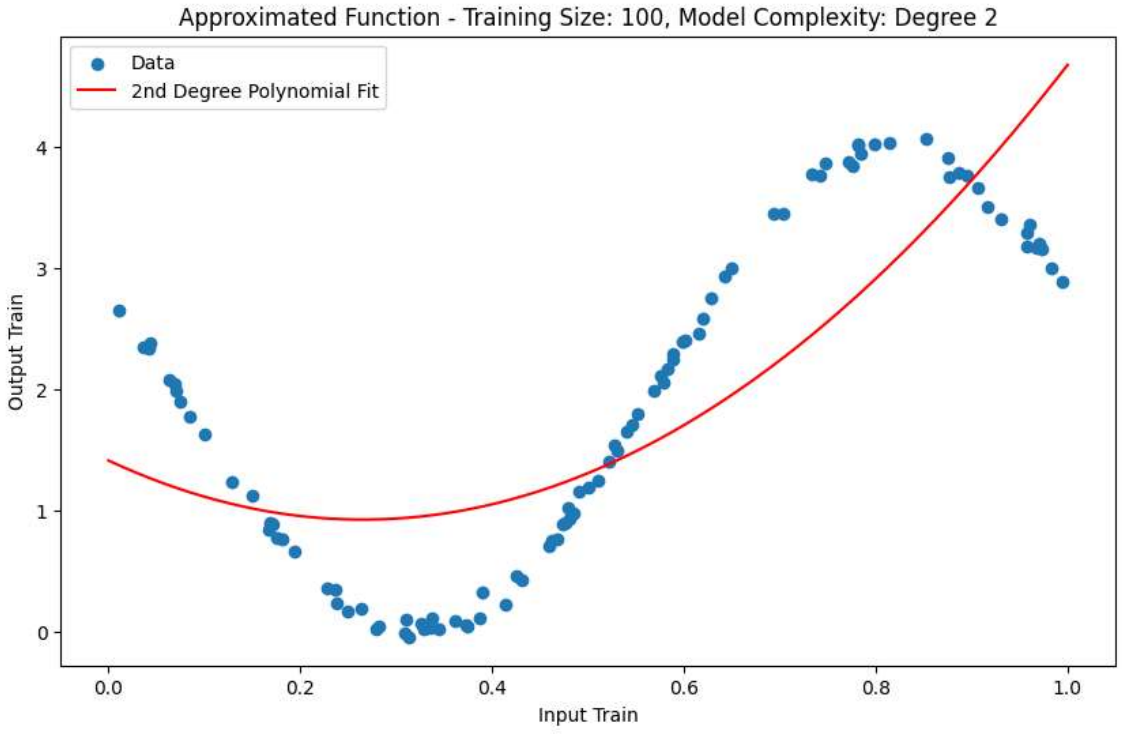


Fig33

- The values of x roughly lie between 0 and 1 (they being included).
- The y values roughly lie between -0.14576 and 4.1118.
- The number of test data points is 300 (whole 30% of 1001).

We can observe from the train and test data that a linear or quadratic polynomial won't be a good fit and we need at least a cubic polynomial or more for a good fit.

1. **For degree/model complexity being 2**



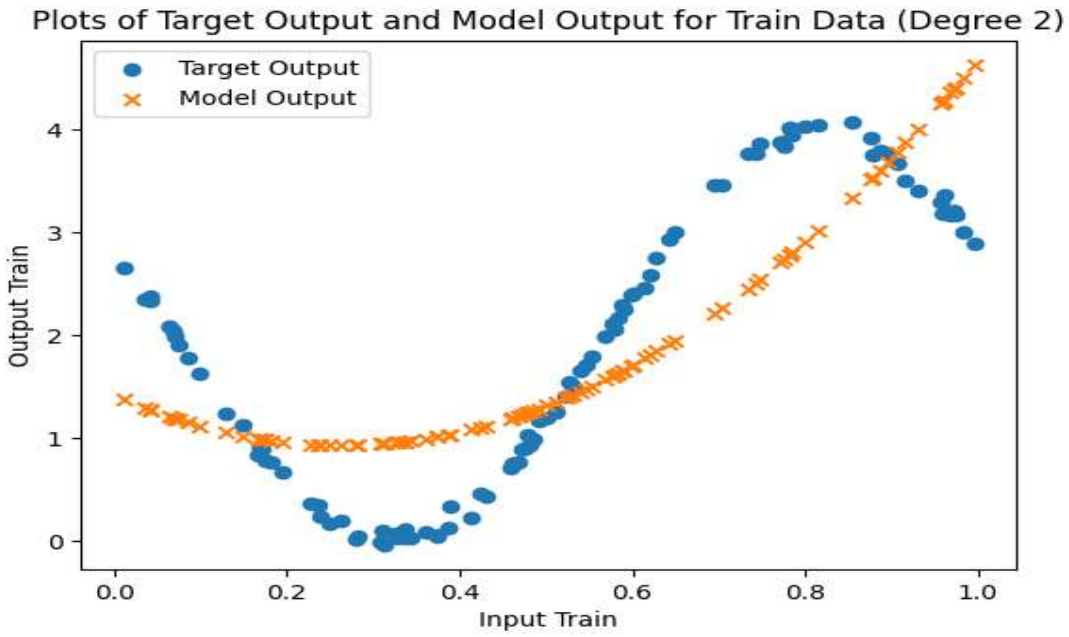


Fig34 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 2)

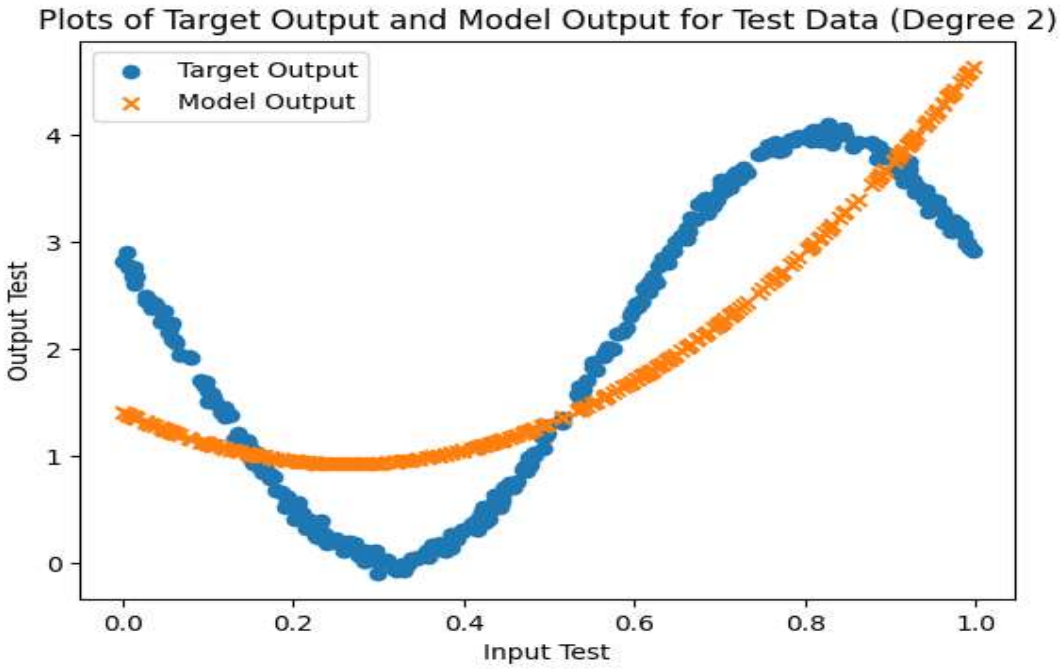


Fig35 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 2)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 2:	1.4116806053698707
w1 value of degree 2:	-3.6801260237518654
w2 value of degree 2:	6.944437390825156
MSE for degree 2 training data:	0.6493146817423969
MSE Percent for degree 2 training data:	35.22669671979363
MSE for degree 2 test data:	0.6766475819083582
MSE Percent for degree 2 test data:	33.830272284303305

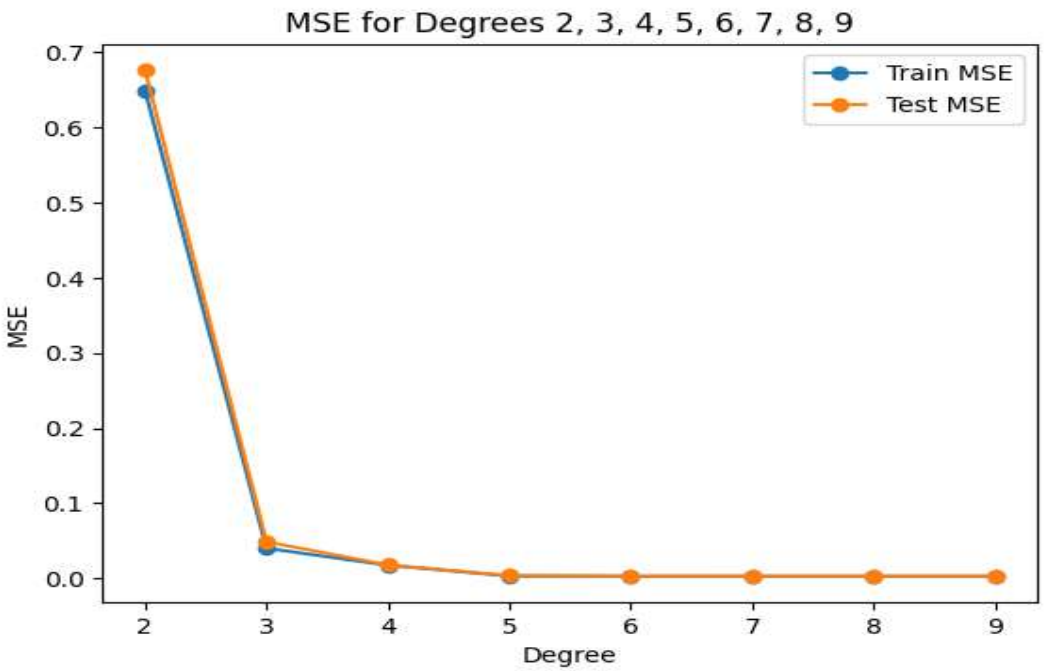


Fig36 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

2. **For degree/model complexity being 3**

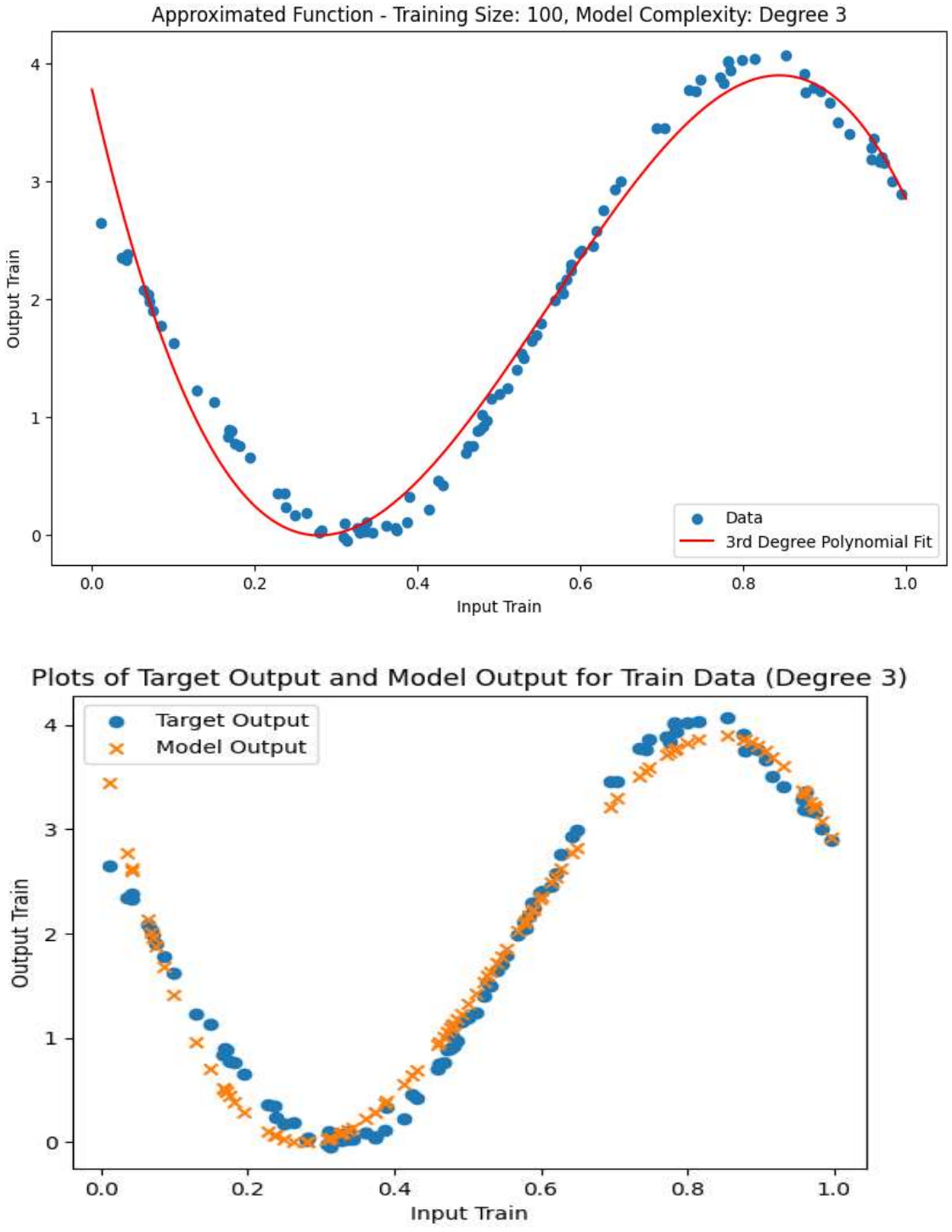


Fig37 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 3)

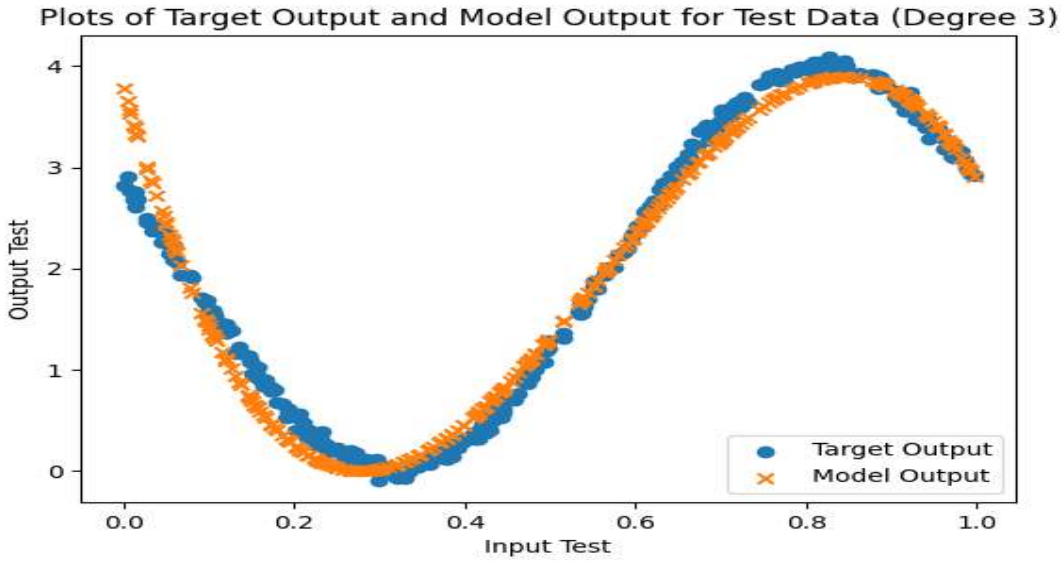


Fig38 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 3).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 3:	3.7758852790534587
w1 value of degree 3:	-30.435076118905577
w2 value of degree 3:	72.58386374766187
w3 value of degree 3:	-43.07264446697374
MSE for degree 3 training data:	0.040134195031423736
MSE Percent for degree 3 training data:	2.1773650838622314
MSE for degree 3 test data:	0.04874611489189096
MSE Percent for degree 3 test data:	2.4371539686045147

We observe that **degree 3 is a better fit than degree 2** and **both train and test MSE have reduced significantly (steep decline)** for instance from **degree 2 to 3 train error has reduced from 0.649 to 0.040** and **for test data test error has reduced from 0.676 to 0.048.**

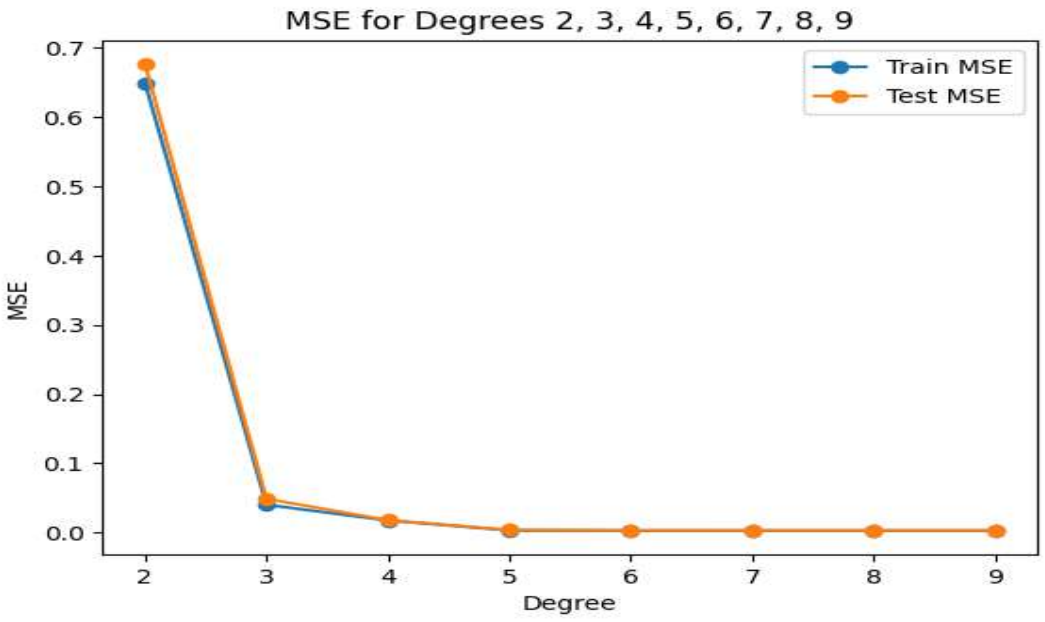
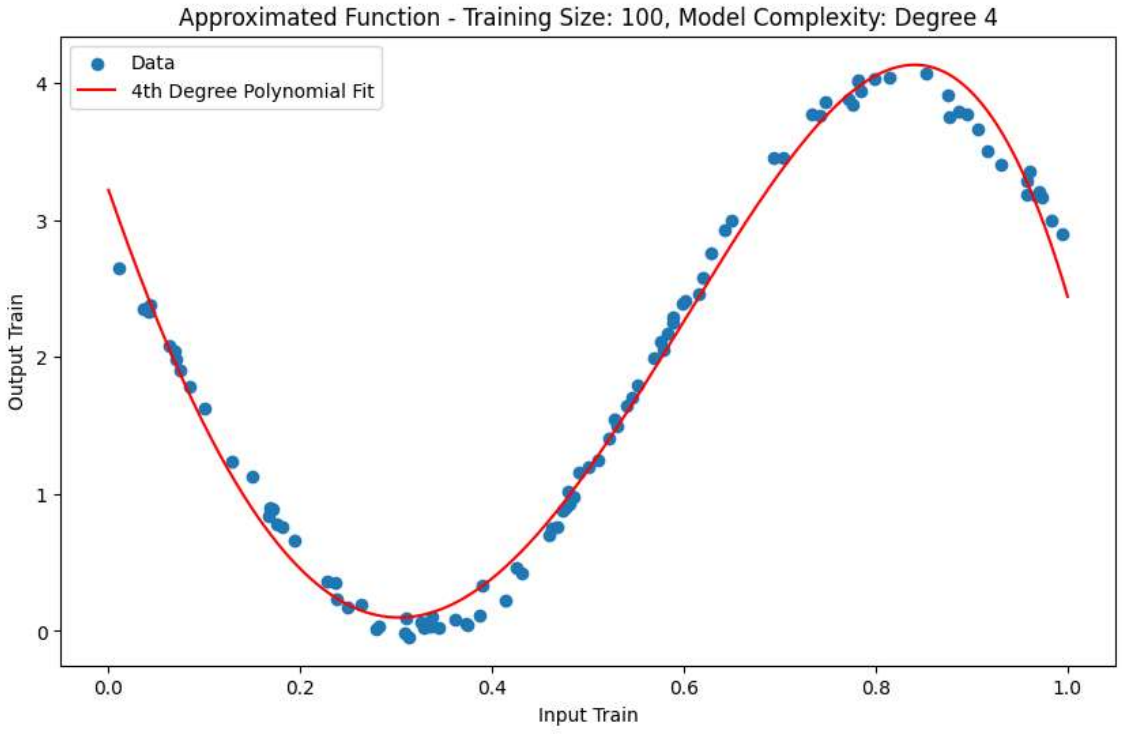


Fig39 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

3. For degree/model complexity being 4



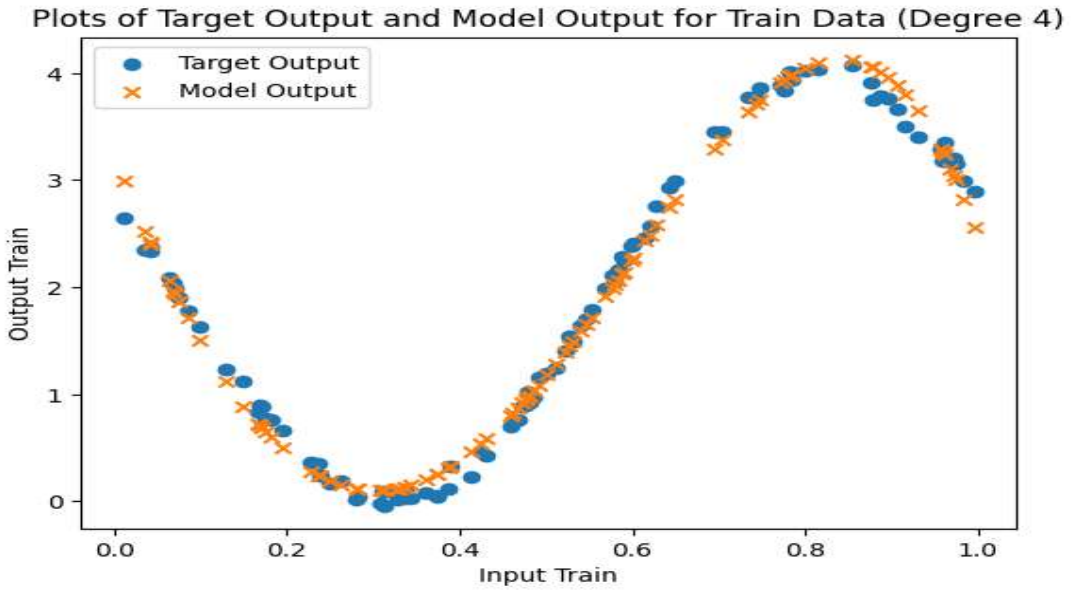


Fig40 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 4)

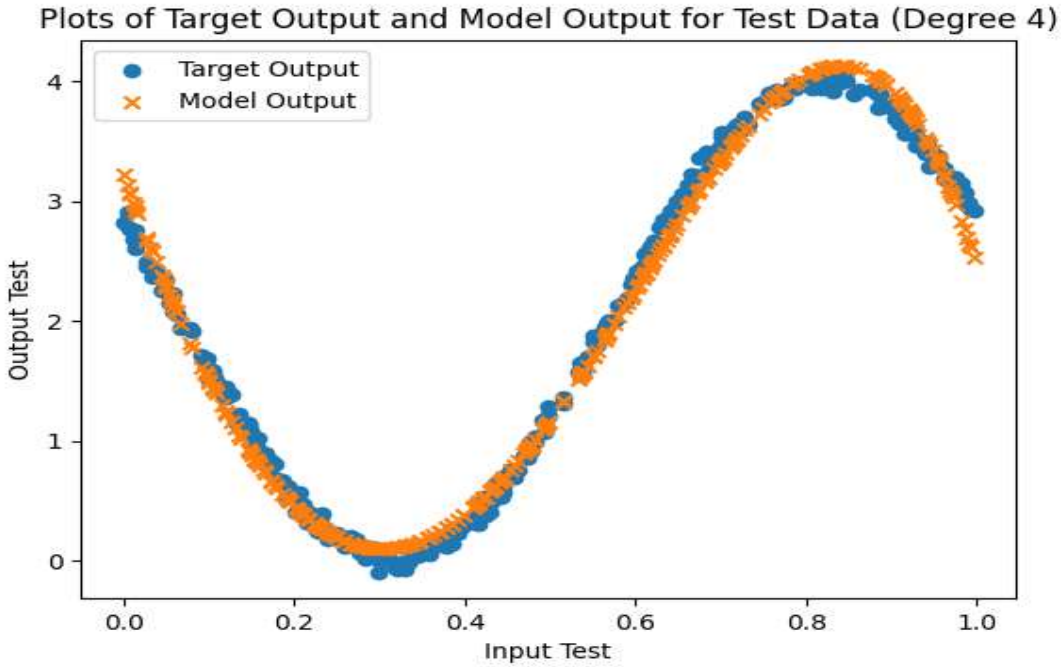


Fig41 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 4).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 4:	3.2182088218921985
w1 value of degree 4:	-20.209910014526148
w2 value of degree 4:	28.357515754163366
w3 value of degree 4:	24.508130166526968
w4 value of degree 4:	-33.43381748200186
MSE for degree 4 training data:	0.017548527963139902
MSE Percent for degree 4 training data:	0.9520448094250806
MSE for degree 4 test data:	0.017868907734674006
MSE Percent for degree 4 test data:	0.8933897500707937

We observe that:-

- Degree 4 is a better fit than degree 3.
- Both train and test MSE have reduced from degree 3 to 4 but not as drastically as they did from degree 2 to 3.

Moving onto 4 degree from 3 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 2 to 3 which is the case as observed from the values of train and test MSE.

This indicates the model has probably become a decent fit for the data and further increase in degrees may not be that beneficial as it may lead to:

- **Overfitting (model may not be able to generalise).**
- **More complexity**.
- **Requirement of more training samples** (the thumb rule mentioned earlier).
- **Very less reduction in train and test MSE values**.
- **Huge coefficient values which in other words means incurring the overhead/cost of regularisation in case of the overfitting.**

Whether these things actually occur in case of train samples being 100 (a sufficient amount even for 9 degree polynomials) is becoming evident as we are proceeding ahead and obtaining results for higher degrees.

A point to observe is that the values of our parameters (w0, w1, w2, w3) are not that extreme, hence apparently there is no need of regularisation as of now.

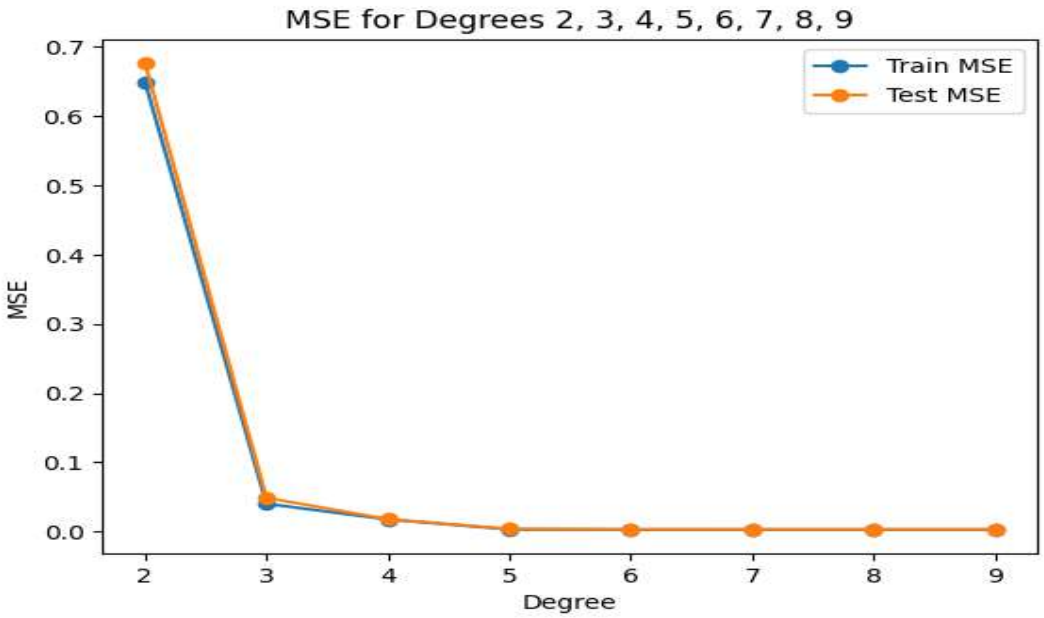
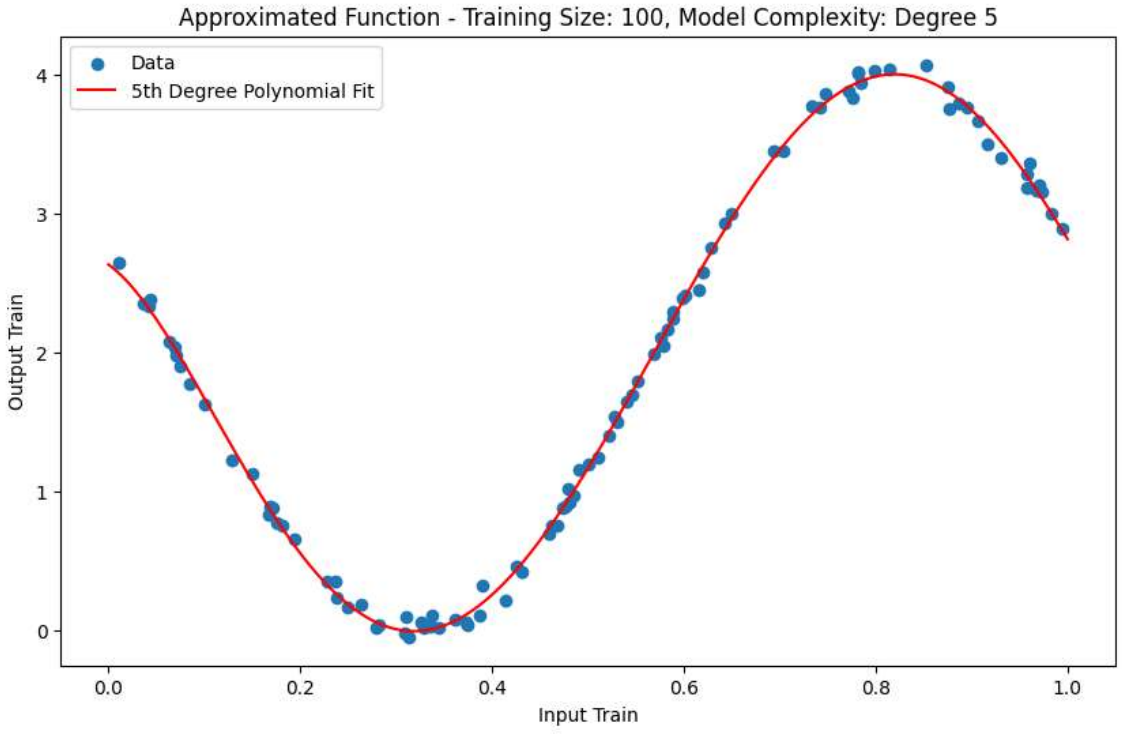


Fig42 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

4. **For degree/model complexity being 5**



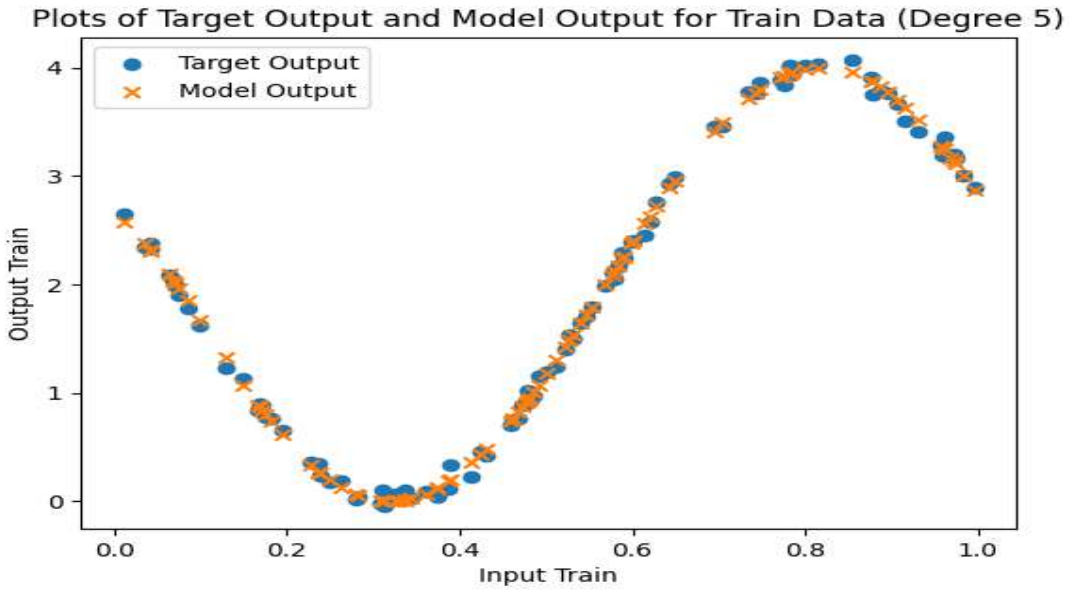


Fig43 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 5)

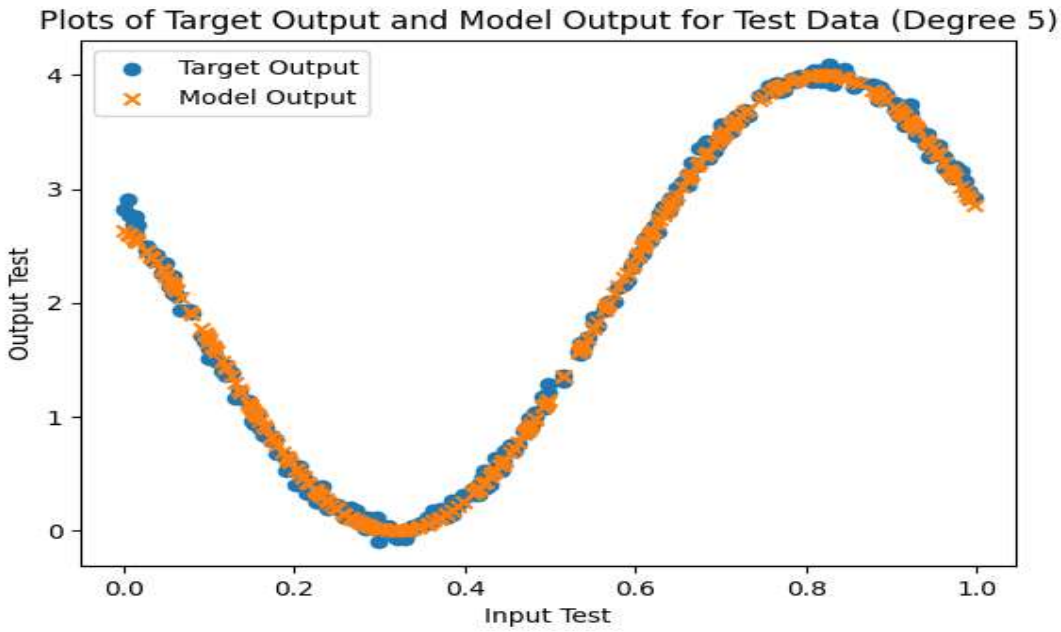


Fig44 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 5).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 5: 2.6335809711185343

w1 value of degree 5: -4.974130630945128

w2 value of degree 5: -71.67641053431761

w3 value of degree 5: 282.1793874774432

w4 value of degree 5: -317.17954468409584

w5 value of degree 5: 111.8328120457395

MSE for degree 5 training data: 0.0031731632785307567

MSE Percent for degree 5 train data: 0.17215082855547625

MSE for degree 5 test data: 0.0037901297527665762

MSE Percent for degree 5 test data: 0.189494686683589

We observe that:-

- Degree 5 is a better fit than degree 4.
- Both train and test MSE have reduced from degree 4 to 5 but not as drastically as they did from degree 3 to 4.

Moving onto 5 degree from 4 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 3 to 4 which is the case as observed from the values of train and test MSE.

This indicates the model has probably become a decent fit for the data and <u>further increase in degrees may not be that beneficial</u> as it may lead to:

- **Overfitting (model may not be able to generalise).**
- **More complexity**.
- **Requirement of more training samples** (the thumb rule mentioned earlier).
- **Very less reduction in train and test MSE values**.
- **Huge coefficient values which in other words means incurring the overhead/cost of regularisation in case of the overfitting.**

The values of our parameters have increased significantly because the number of training samples are limited.

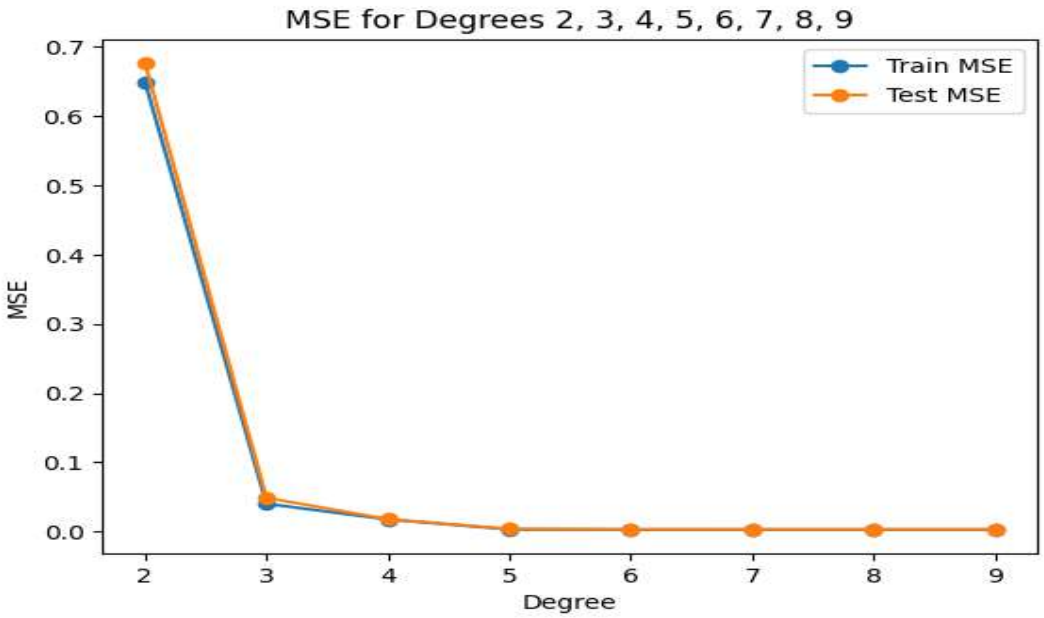
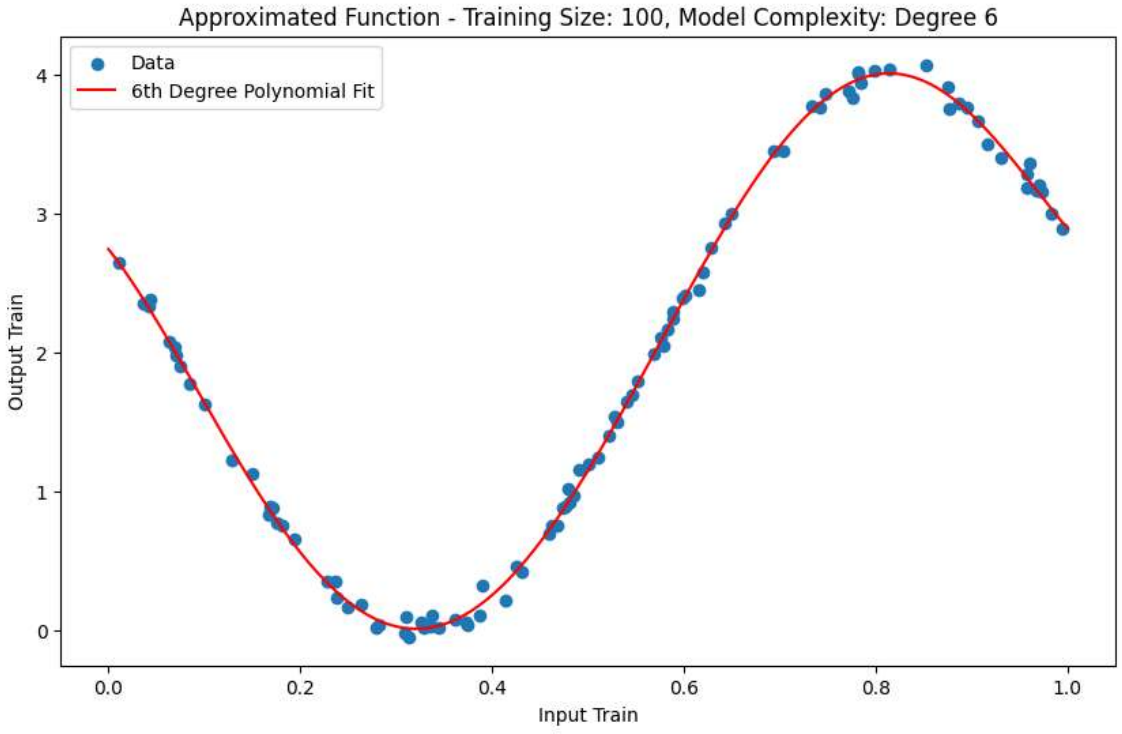


Fig45 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

5. **For degree/model complexity being 6**



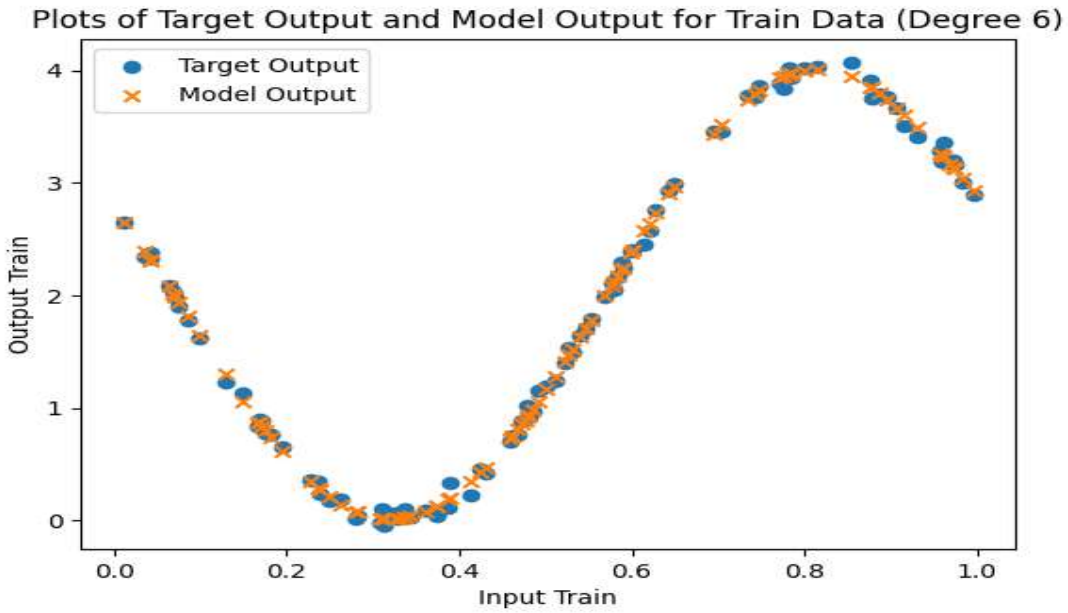


Fig46 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 6)

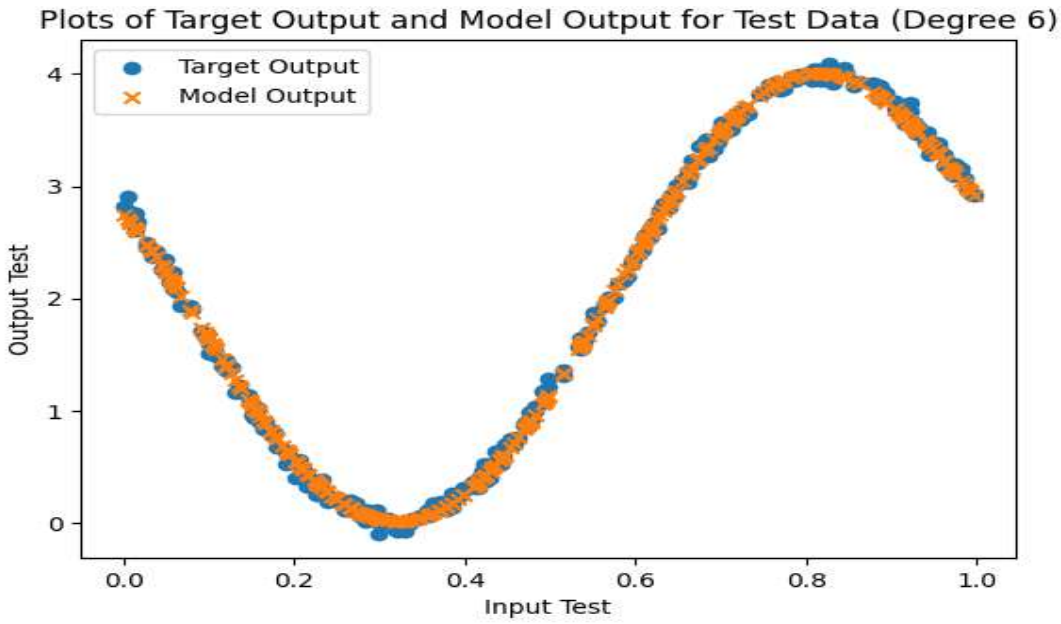


Fig47 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 6).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 6: 2.7441771579082888

w1 value of degree 6: -8.758997718636007

w2 value of degree 6: -36.69883947746874

w3 value of degree 6: 147.34466124063806

w4 value of degree 6: -68.87611675311524

w5 value of degree 6: -104.74114917226275

w6 value of degree 6: 71.8781089697295

MSE for degree 6 training data: 0.002790271529460346

MSE Percent for degree 6 training data: 0.1513781402114819

MSE for degree 6 test data: 0.0033218013439690504

MSE Percent for degree 6 test data: 0.16607972443187935

We observe that:-

- Degree 6 is a better fit than degree 5.
- Both train and test MSE have reduced from degree 5 to 6 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.

Moving onto 6 degree from 5 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 4 to 5 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have not increased significantly from degree 5 to 6** and the **error has declined but it is significantly less** which means **we are making our model more complex, it is losing its generalisation ability and the coefficients might soon be large at any point so we should select some previous degree model as a suitable one.**

That being said, our test MSE is still low since we have a decent number of training samples but the model is still losing its generalisation ability.

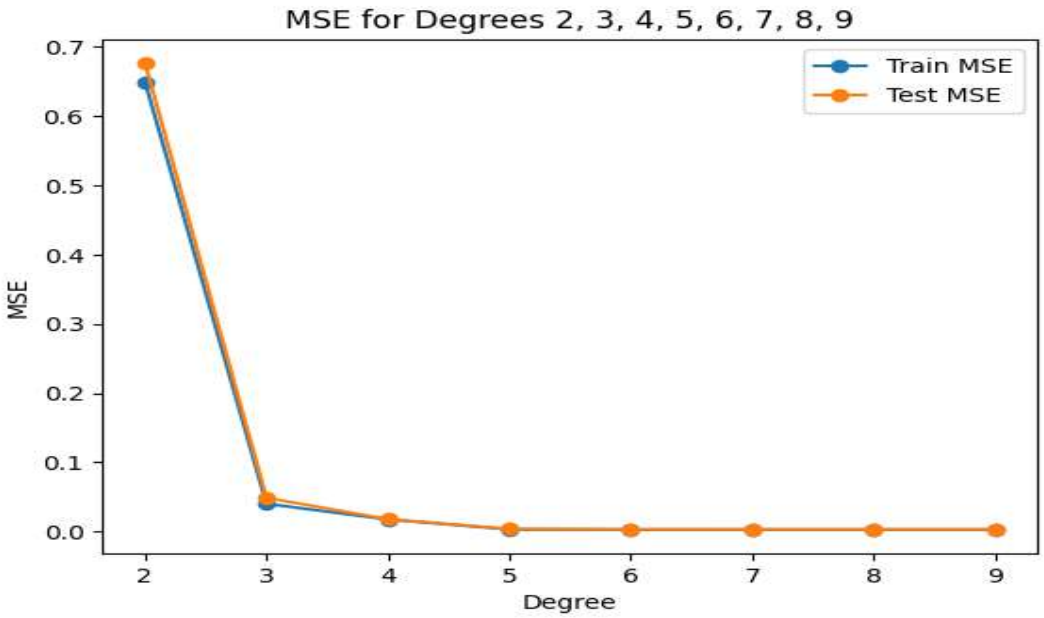


Fig48 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

9) For degree/model complexity being 7

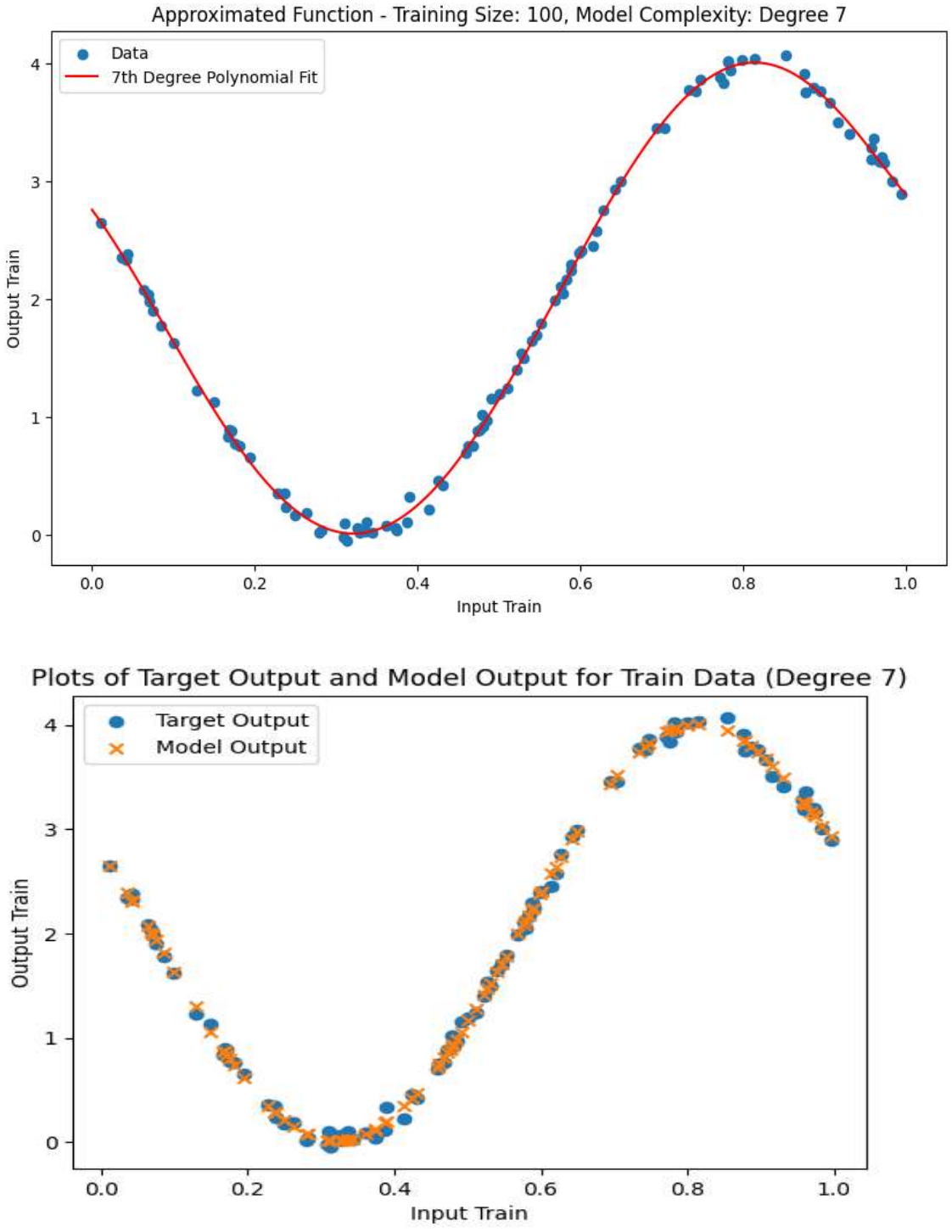


Fig49 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 7)

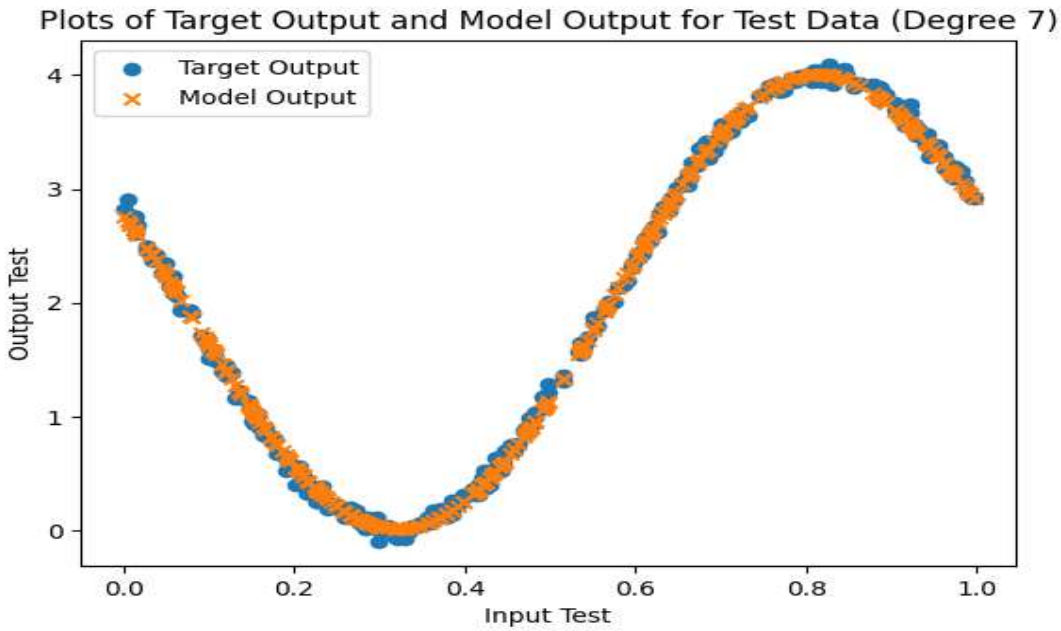


Fig50 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 7).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 7:	2.757586049880186
w1 value of degree 7:	-9.340804601474417
w2 value of degree 7:	-29.547237437712653
w3 value of degree 7:	109.53480686034413
w4 value of degree 7:	31.977656367052077
w5 value of degree 7:	-247.05157414184214
w6 value of degree 7:	173.20745664656351
w7 value of degree 7:	-28.65431541609655
MSE for degree 7 training data:	0.0027862754253993333
MSE Percent for degree 7 training data:	0.15116134310250498
MSE for degree 7 test data:	0.003282966320831845
MSE Percent for degree 7 test data:	0.16413809419181621

We observe that:-

- Degree 7 is a better fit than degree 6.
- Both train and test MSE have reduced from degree 6 to 7 but not as drastically as they did from degree 5 to 6 and the reduction is extremely minute.
- Degree 6 to 7 train error has almost stayed the same.
- For test data, test error has almost stayed the same.

Moving onto 7 degree from 6 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 5 to 6 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased from degree 6 to 7** and the **MSE values have stayed the same so we should have stopped earlier/ at degree 5 or 4 as discussed earlier (since degree 5 seems to have less generalisation capacity as compared to degree 4).**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and train MSE and the model most probably has now lost its generalisation ability.

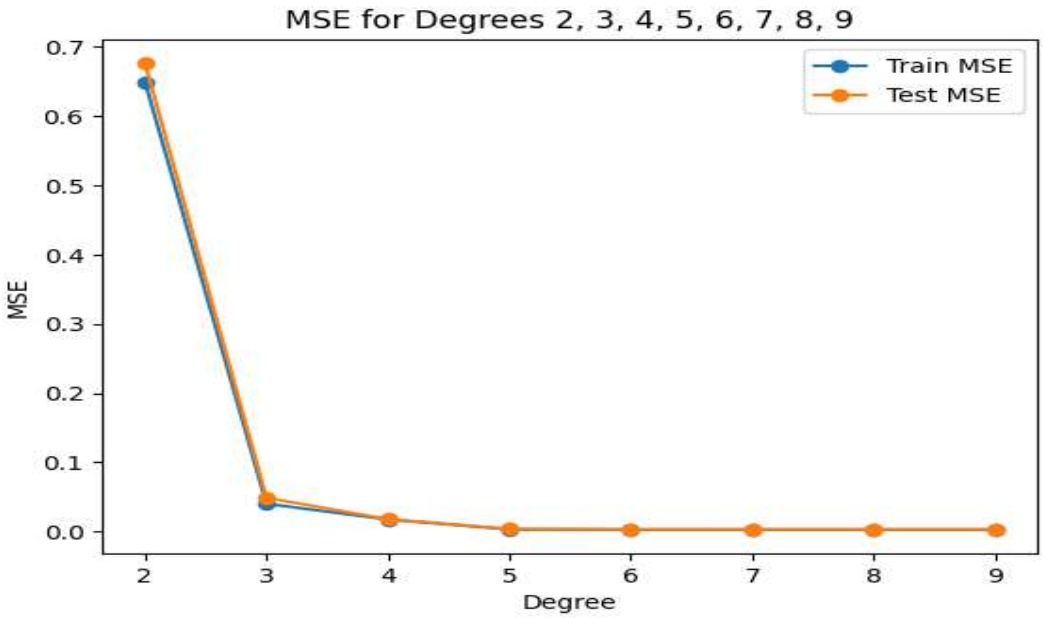


Fig51 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing complexity of the model which may result in huge values of parameters, but since we have just sufficient number of training samples so in theory we should be able to observe no increase in test MSE.

And as we see that is the case for results of degrees we see ahead.

10) For degree/model complexity being 8

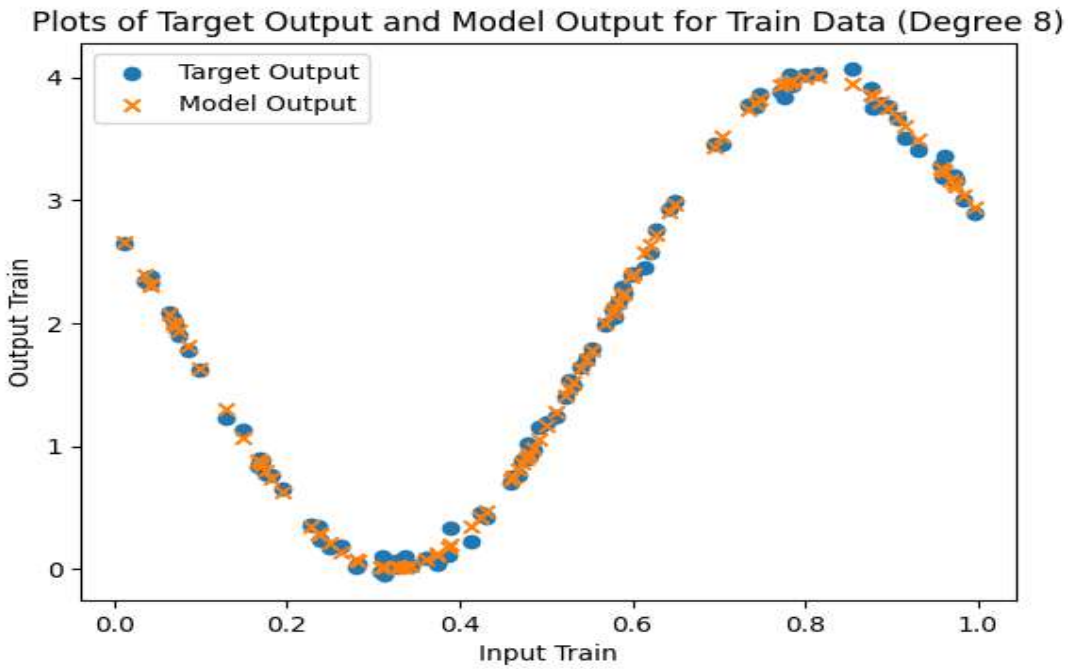


Fig52 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 8)

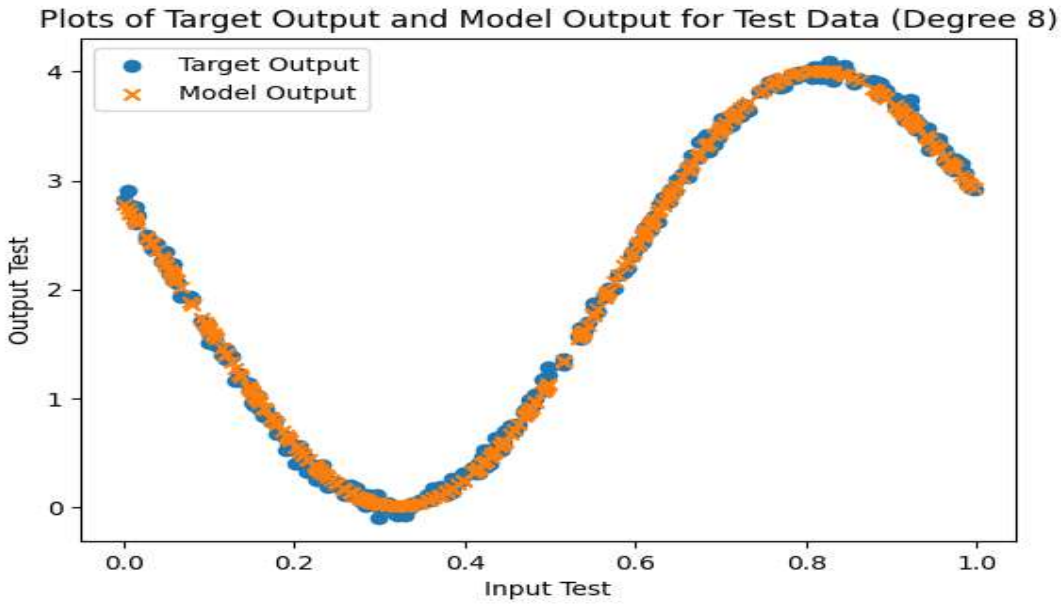


Fig53 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 8)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 8: 2.7825199041341415

w1 value of degree 8: -10.722882259657823

w2 value of degree 8: -7.157912687013436

w3 value of degree 8: -49.28394762964854

w4 value of degree 8: 618.1642387482934

w5 value of degree 8: -1457.1714936869685

w6 value of degree 8: 1580.3908736091762

w7 value of degree 8: -889.2553172016851

w8 value of degree 8: 215.15583061662983

MSE for degree 8 training data: 0.0027724065859108383

MSE Percent for degree 8 train data: 0.1504089291863346

MSE for degree 8 test data: 0.0032355180554722715

MSE Percent for degree 8 test data: 0.16176582865884095

We observe that:-

- Degree 8 is a better fit than degree 7.
- Both train and test MSE have reduced from degree 7 to 8 but not as drastically as they did from degree 6 to 7 and the reduction is extremely minute.

Moving onto 8 degree from 7 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 6 to 7 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 7 to 8** and the **MSE values have stayed the same.**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and training MSE and the model most probably has now completely lost its generalisation ability.

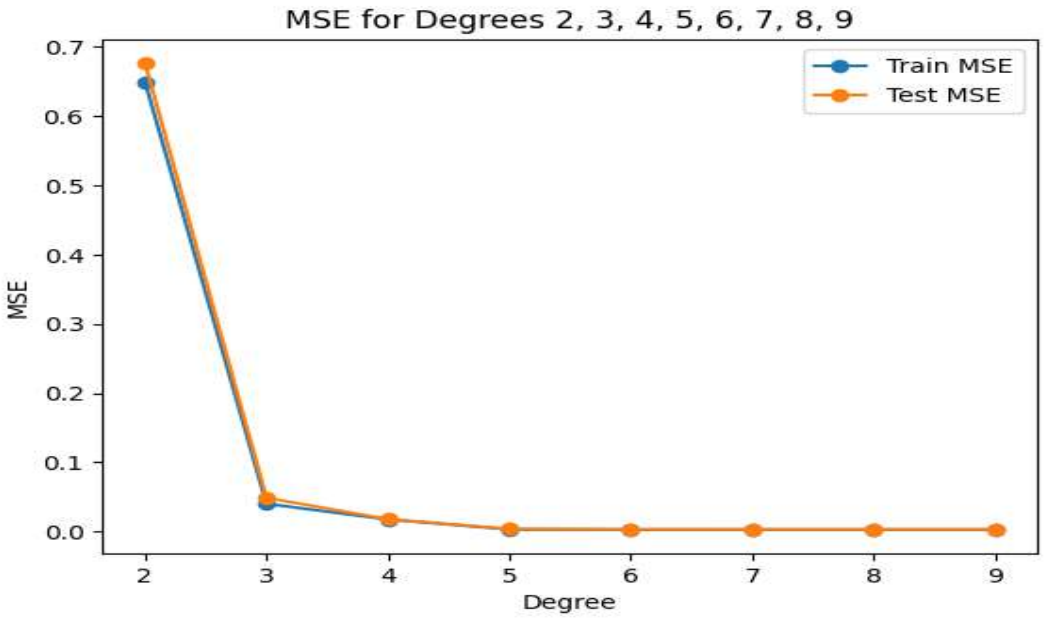


Fig54 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which is resulting in huge values of parameters.

And as we see that is the case for results of degrees we see ahead.

11) For degree/model complexity being 9

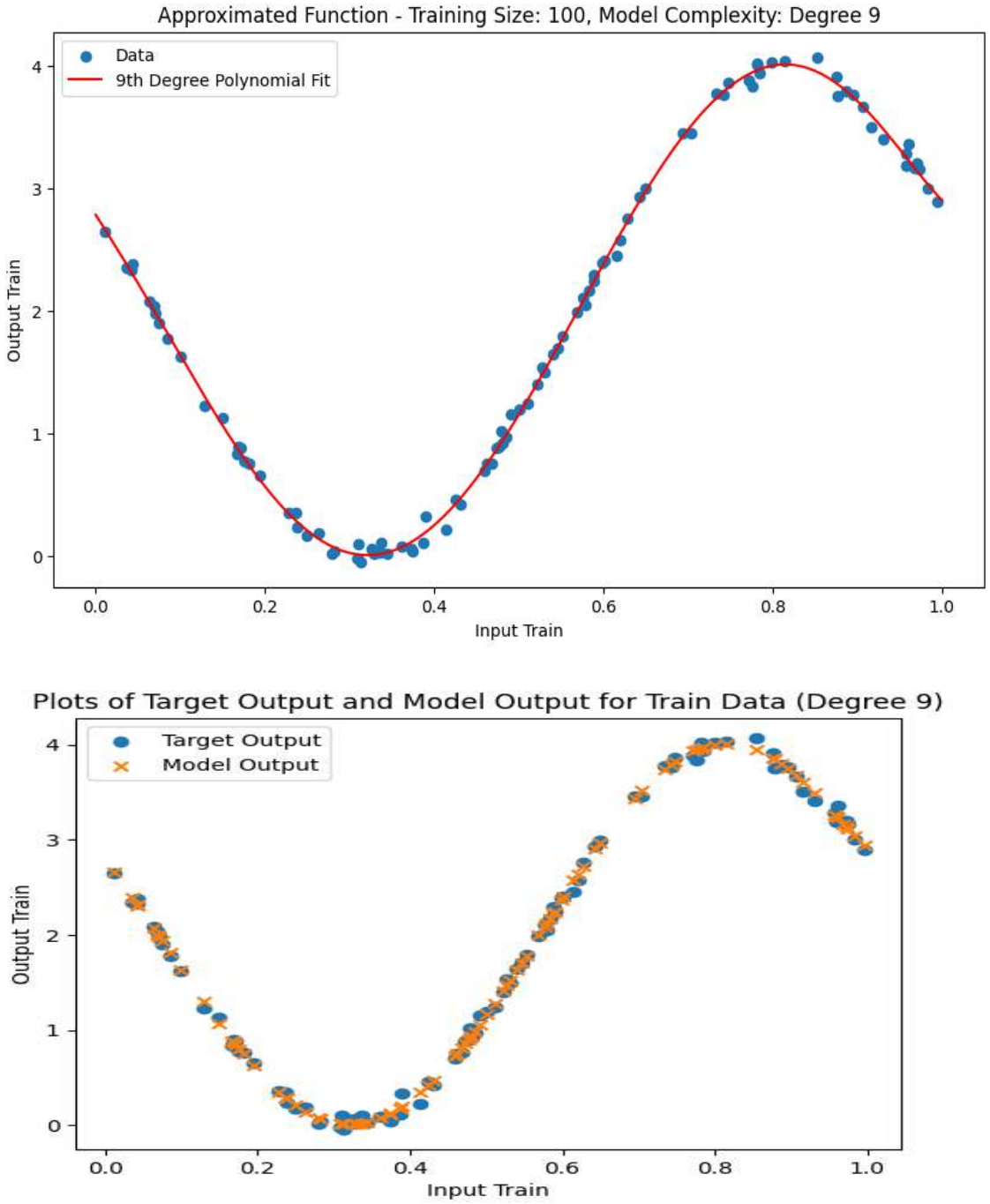


Fig55 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 9)

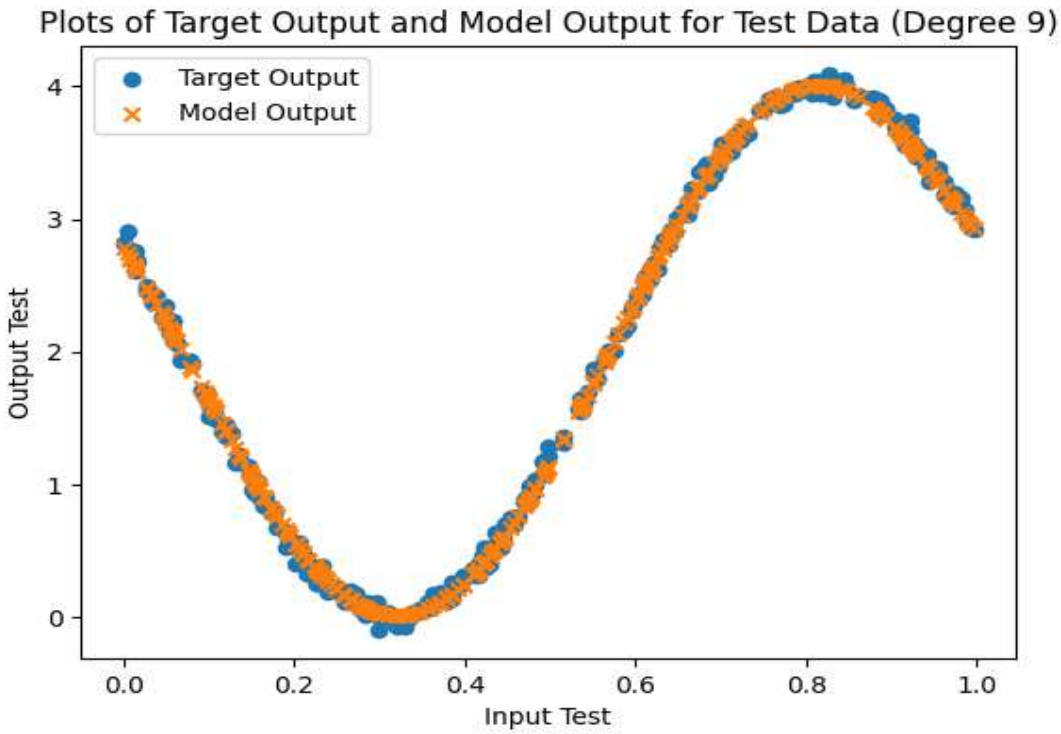


Fig56 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 9)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 9: 2.783961048048033

w1 value of degree 9: -10.816381586562628

w2 value of degree 9: -5.228104293844581

w3 value of degree 9: -66.81441529082076

w4 value of degree 9: 702.3066048553883

w5 value of degree 9: -1690.5418798775063

w6 value of degree 9: 1967.0945062329993

w7 value of degree 9: -1266.787904594501

w8 value of degree 9: 415.3646087781526

w9 value of degree 9: -44.46037055510169

MSE for degree 9 training data: 0.0027723856341162136

MSE Percent for degree 9 train data: 0.15040779250710073

MSE for degree 9 test data: 0.0032365423107320944

MSE Percent for degree 9 test data: 0.16181703823270913

We observe that:-

- Degree 9 is a better fit than degree 8.
- Both train and test MSE have reduced from degree 8 to 9 but not as drastically as they did from degree 7 to 8 and the reduction is extremely minute.
- Degree 8 to 9 train error has stayed the same.
- For test data, test error has stayed the same.

Moving onto 9 degree from 8 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 7 to 8 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 8 to 9** and the **MSE values have stayed the same.**

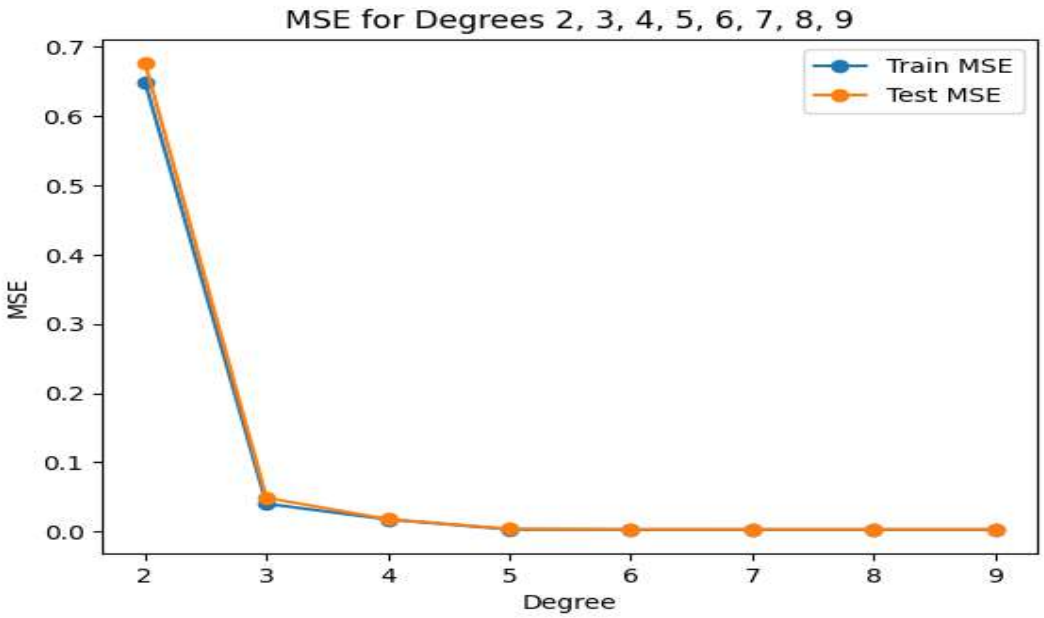


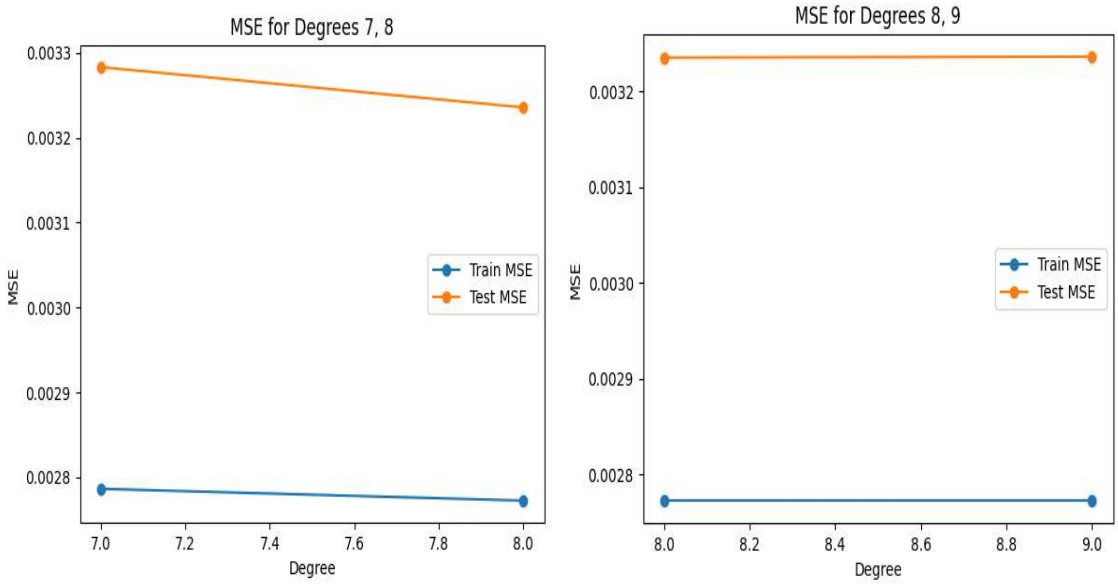
Fig57 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which is resulting in huge values of parameters.

Regularization:

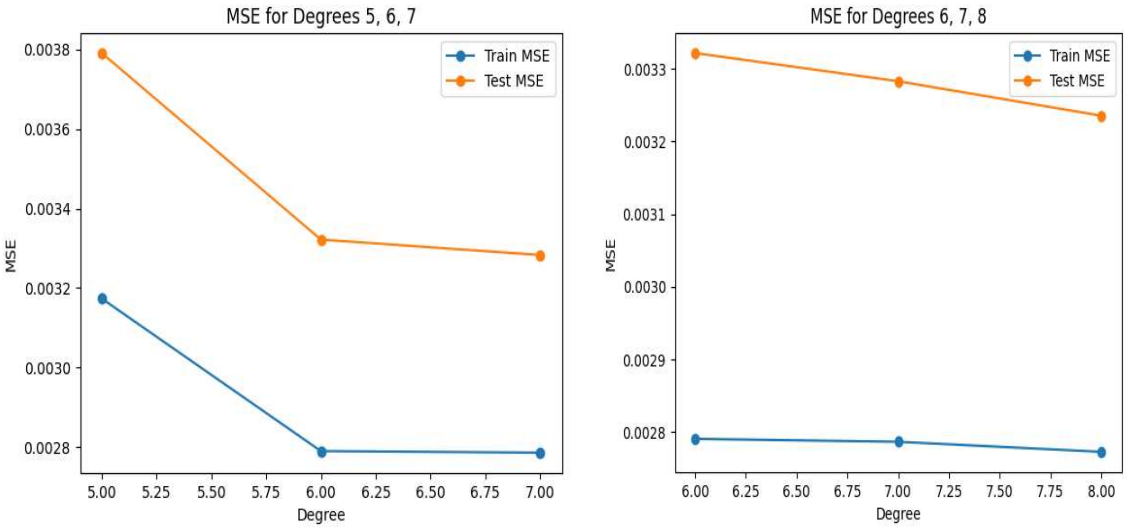
For regularization we take a close look at our MSE plots but we do it piecewise.

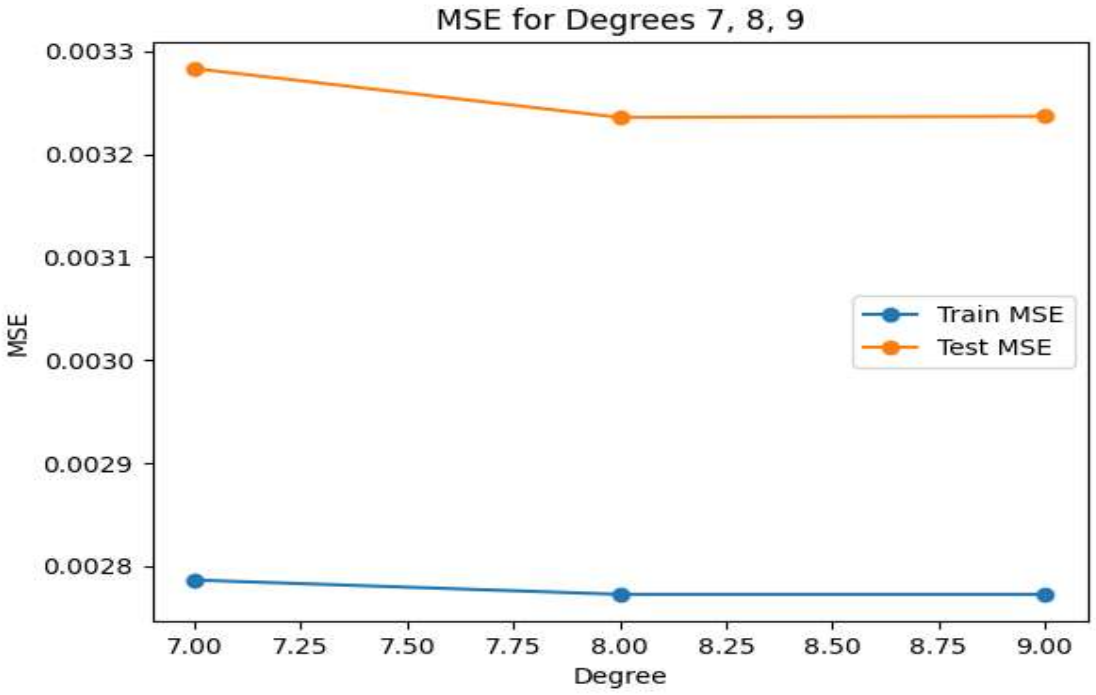
For 100 samples we have few piecewise MSE curves



If we just see the graphs of MSE for degrees 6, 7 then 7, 8 and finally 8, 9 we observe an almost no increase in test MSE from 8 to 9 whether it is actually true can be observed as we plot MSE for 3, 4, 5, 6 and 7 degrees together.

MSE curves for 3 degrees combined

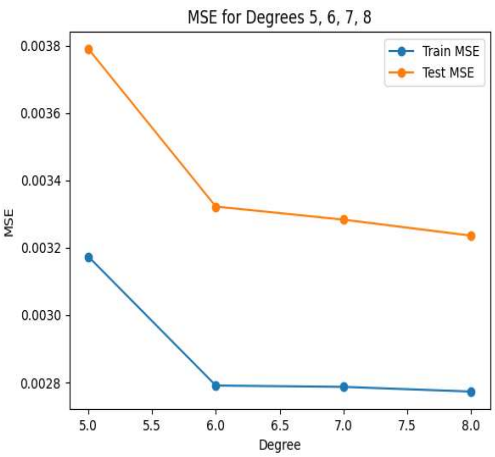


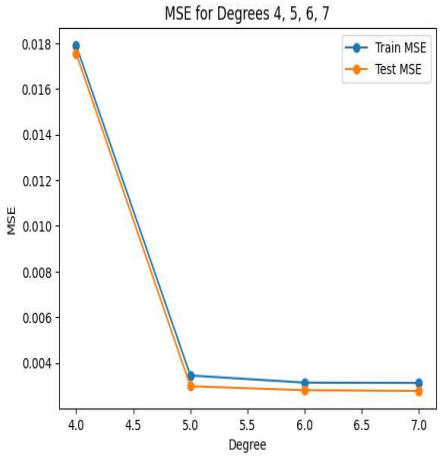


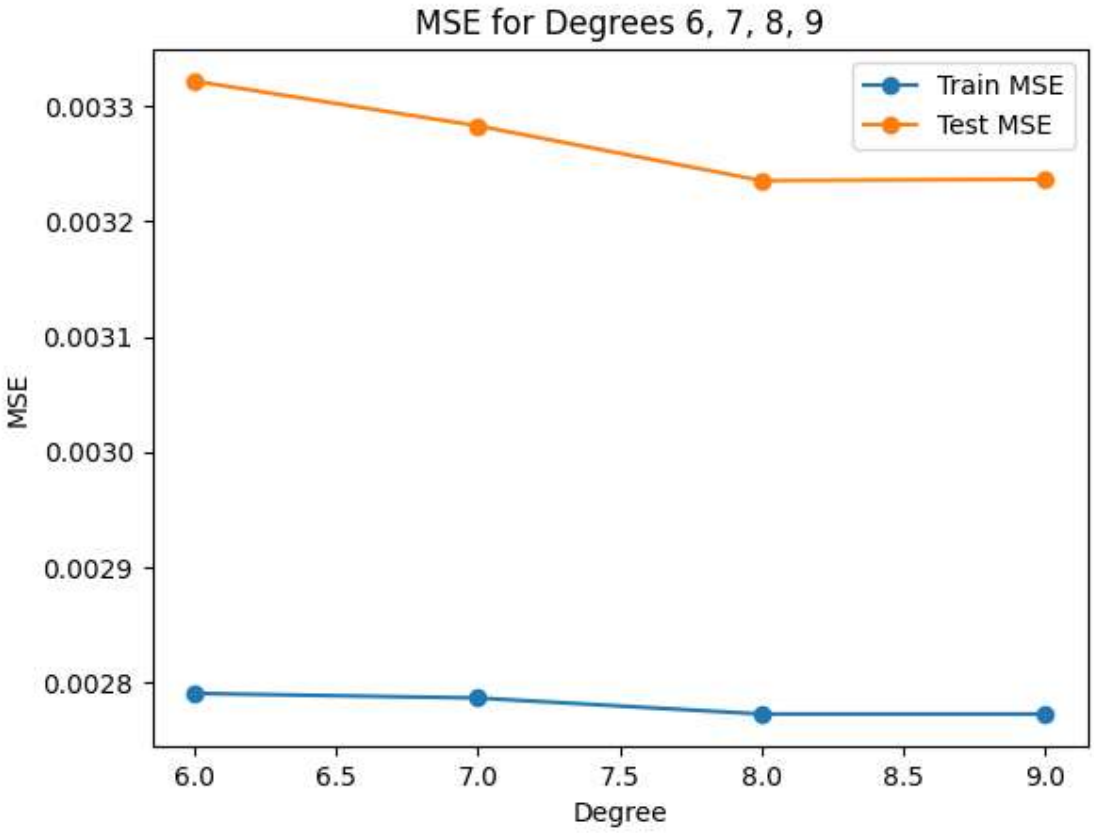
While plotting triplets as well we observe that we have obtained no increase in the test MSE from degree 8 to 9.

Let us continue observing it further.

MSE curves for 4 degrees combined. This time we plot all the possible combinations of 4 degrees (in series).



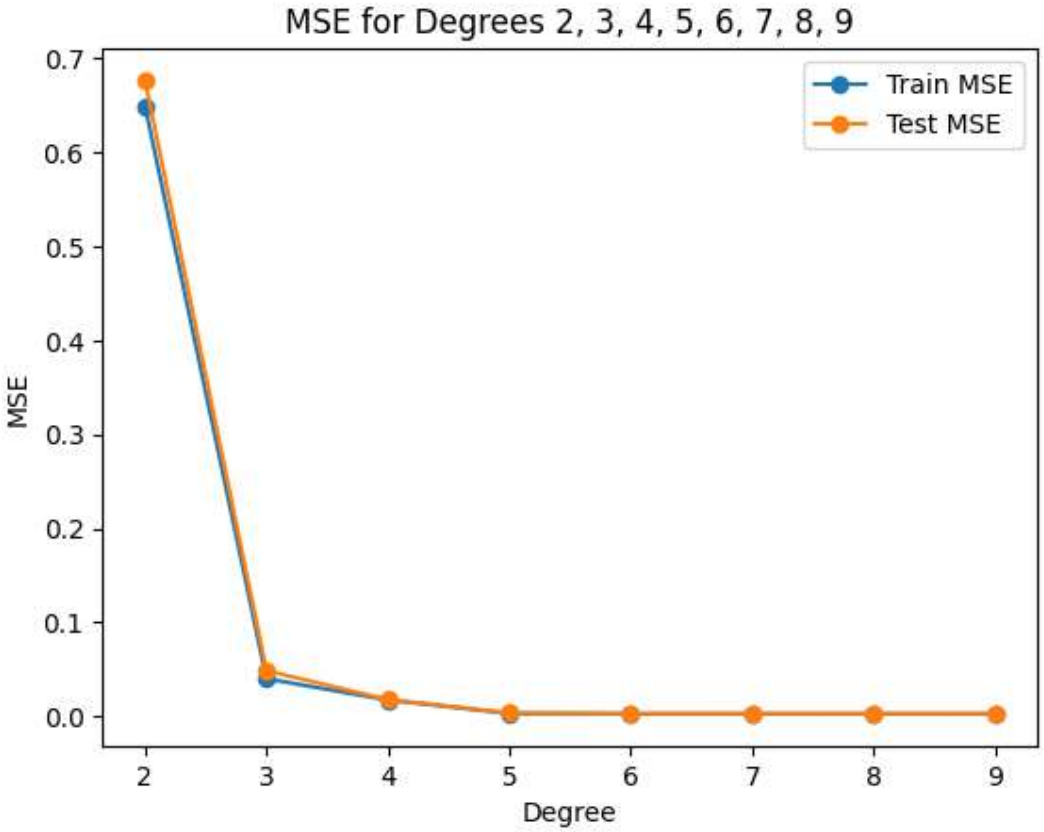




What we observe still is that our test MSE is not increasing from 8 to 9 degree.

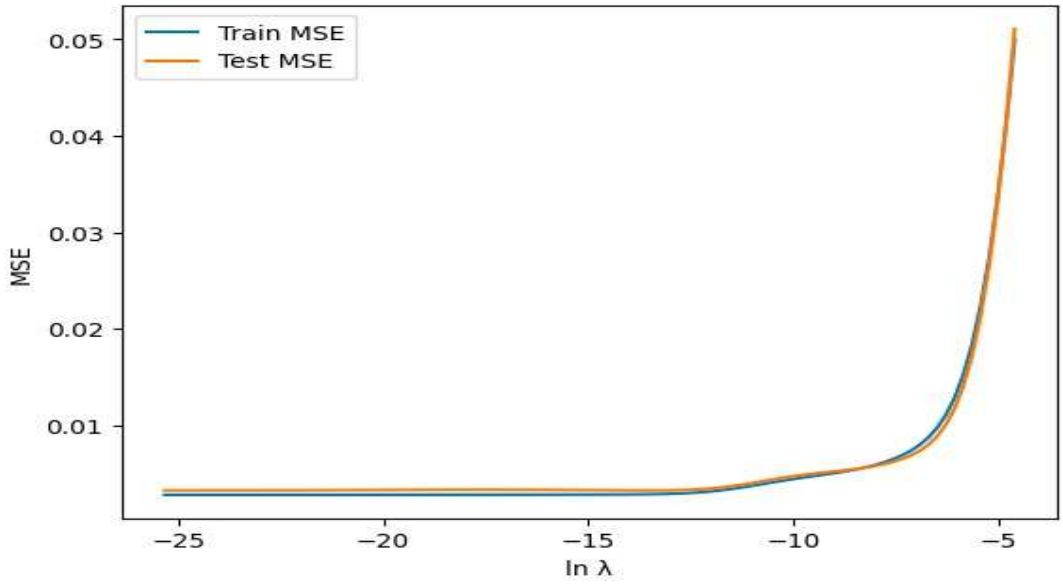
Let us keep on observing.

Cumulative MSE plot



Now we perform regularization for degree 9 for the case of 100 train samples and 300 test sample.

Regularization for degree 9 when number of train samples is 100 and test samples is 300



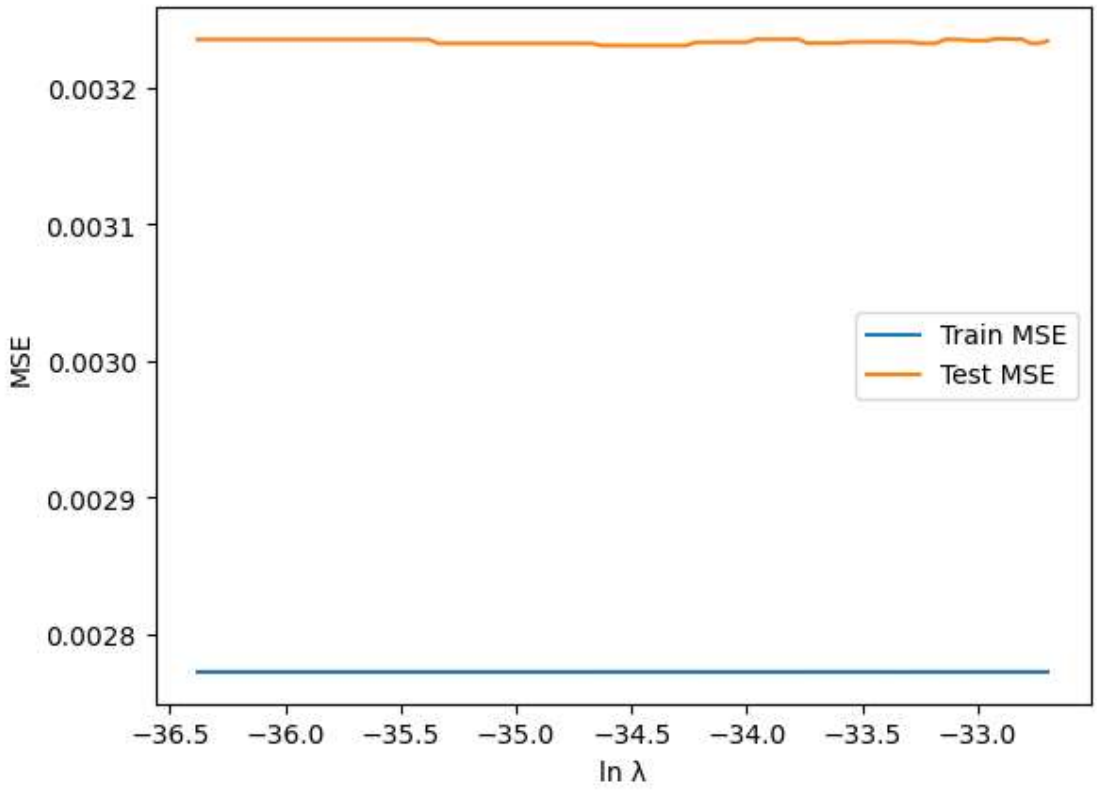
Now we select values of that lambda where test MSE is the lowest.

```
print(mse_test[80])
print(np.argmin(mse_train))
print(np.argmin(mse_test))
print(mse_train[47])
print(min(mse_test))
print(alpha_values[47])
print(alpha_values[80])
```

0.0027723707011414423
0.003233415880064957
80
47
0.0027723906567251183
0.0032307933304735604
9.111627561154906e-16
3.111209762773268e-15

We run a code to find that value of lambda where our test MSE is minimum.

Here is that particular graph which has helped us find the global minima of test MSE for graph of ln lambda VS train and test MSE.



Essentially both train and test MSE for a certain value of lambda while regularising is lower than that of train and test MSE values without regularisation.

After regularization we obtain the following plot for **lambda**=3.111209762773268e-15

Before Regularisation

w0 value of degree 9: 2.783961048048033
w1 value of degree 9: -10.816381586562628
w2 value of degree 9: -5.228104293844581
w3 value of degree 9: -66.81441529082076
w4 value of degree 9: 702.3066048553883
w5 value of degree 9: -1690.5418798775063
w6 value of degree 9: 1967.0945062329993
w7 value of degree 9: -1266.787904594501
w8 value of degree 9: 415.3646087781526
w9 value of degree 9: -44.46037055510169

After regularization

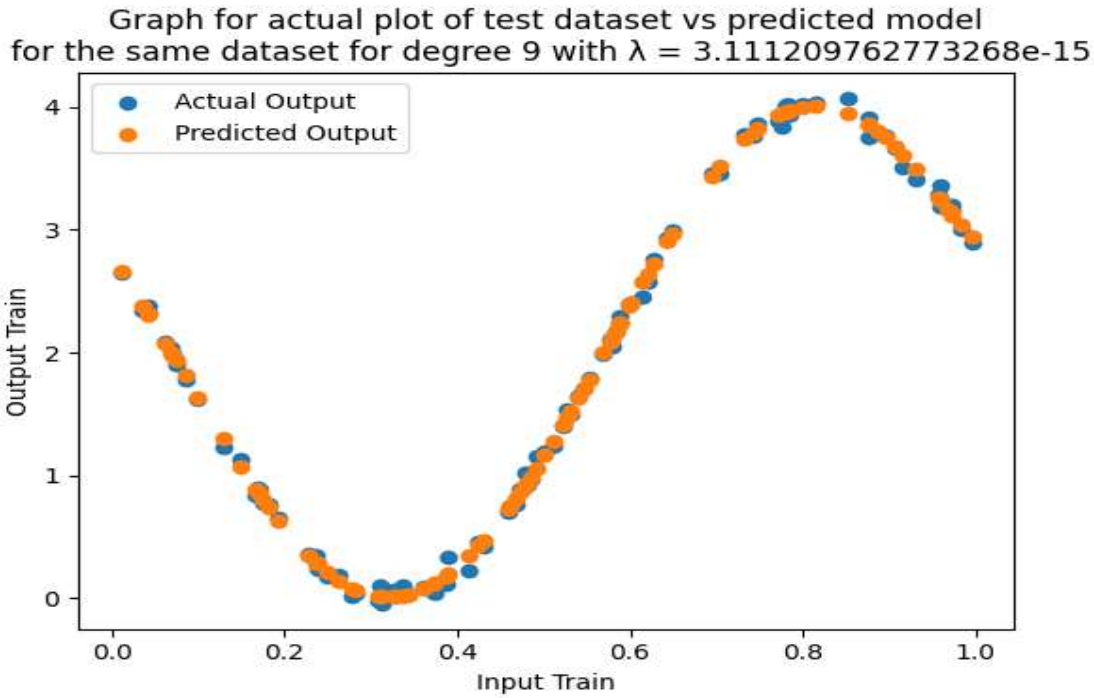
w0 value of degree 9 2.7837318502807453
w1 value of degree 9 -10.81603641745916
w2 value of degree 9 -5.2282209487293585
w3 value of degree 9 -66.7823981640795
w4 value of degree 9 702.1170120533789
w5 value of degree 9 -1690.0066771165002
w6 value of degree 9 1966.217372402607
w7 value of degree 9 -1265.942660772067
w8 value of degree 9 414.9217853435839
w9 value of degree 9 -44.3630640352421

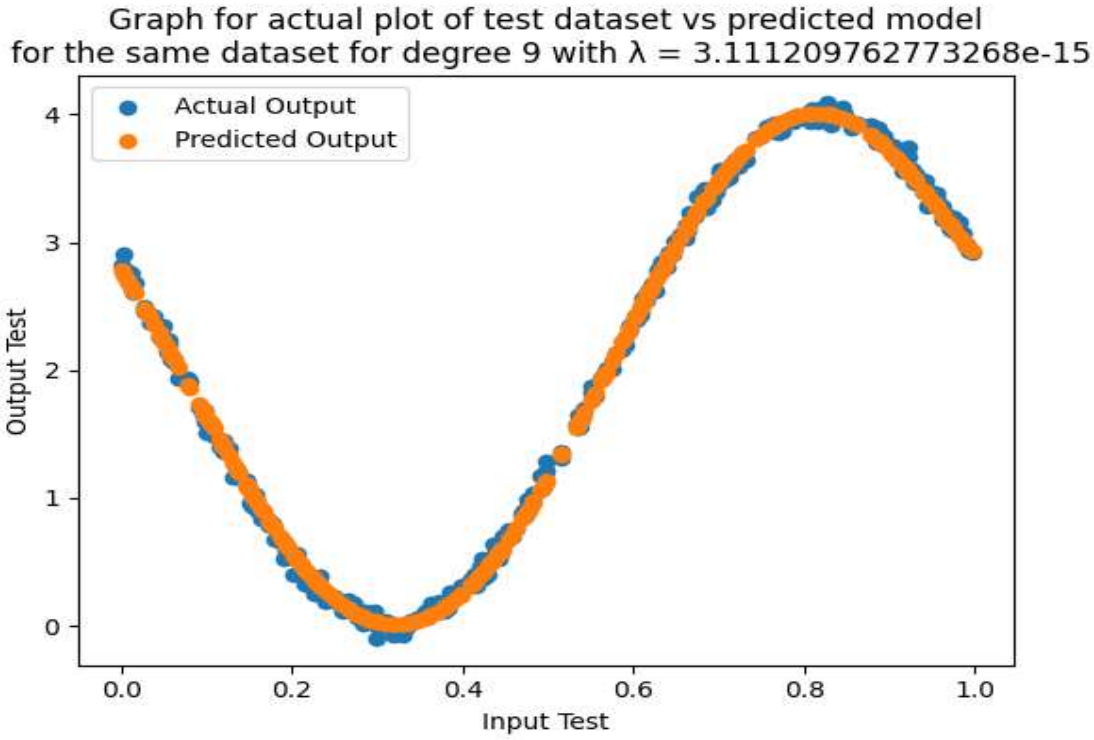
Before Regularisation:

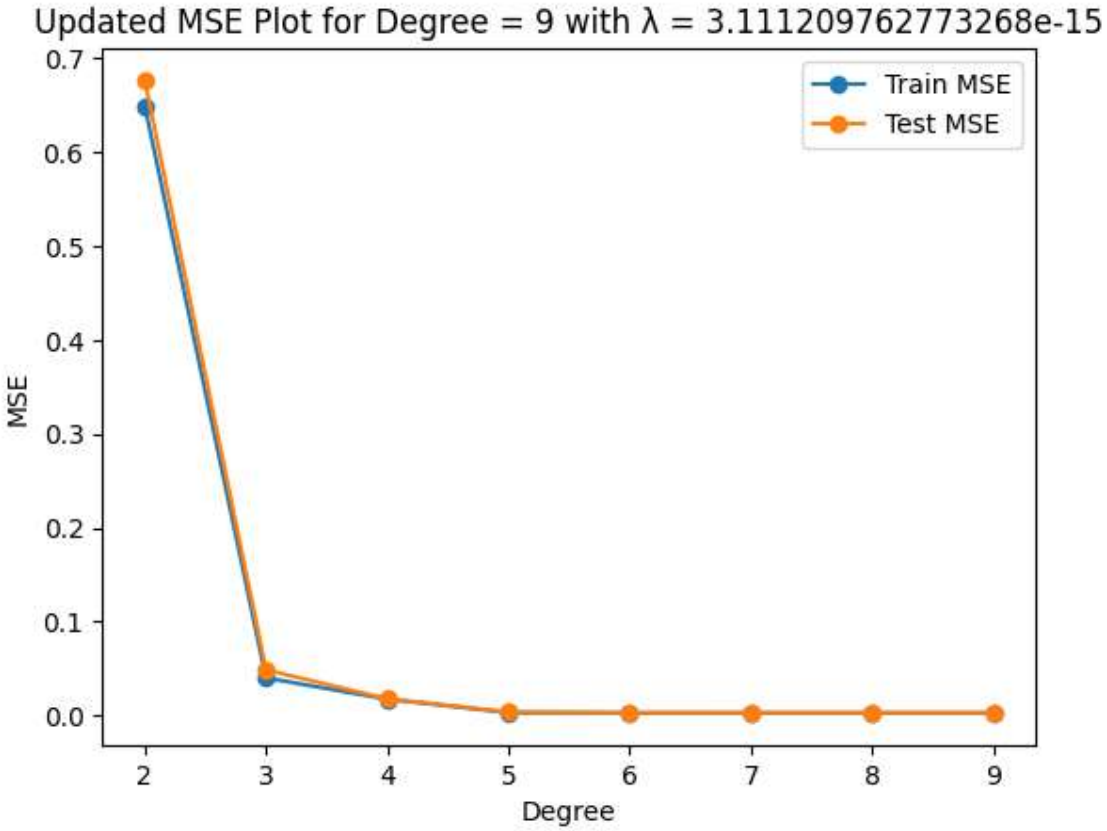
MSE for degree 9 training data: 0.0027723856341162136
MSE Percent for degree 9 training data: 0.15040779250710073
MSE for degree 9 test data: 0.0032365423107320944
MSE Percent for degree 9 test data: 0.16181703823270913

After Regularisation:

MSE for degree 9 training data: 0.0027723707011414423
MSE Percent for degree 9 training data: 0.15040698236159167
MSE for degree 9 test data: 0.003233415880064957
MSE Percent for degree 9 test data: 0.16166072643381232


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 3.111209762773268e-15
Actual Output
Predicted Output
Output Train
Input Train
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 3.111209762773268e-15
Actual Output
Predicted Output
Output Test
Input Test
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0



If we zoom it and only see the graph for 7 and 8, 8 and 9 degrees we will observe that the test MSE does not increase from 8 to 9 after regularisation.

Note: When we consider 50 train samples considered for 50 train samples, those 50 train samples must in turn have 10 train samples considered for the case of 10 train samples.

Observations for 50 samples:

Plot for train data which is 50 train samples.

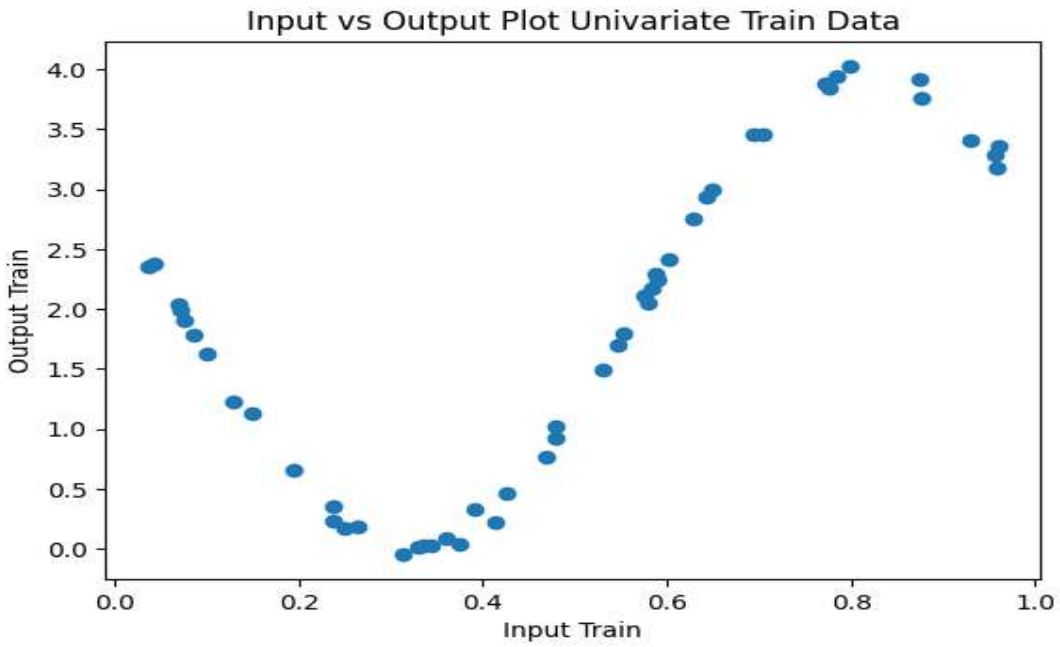


Fig58

- The values of x are between 0.03 and 0.96 (they being included).
- The y values should roughly lie between -0.11 and 4.01 The number of train data points is 50.
- The number of test samples is 300.

Plot of the test data (300 samples) for 50 train samples case.

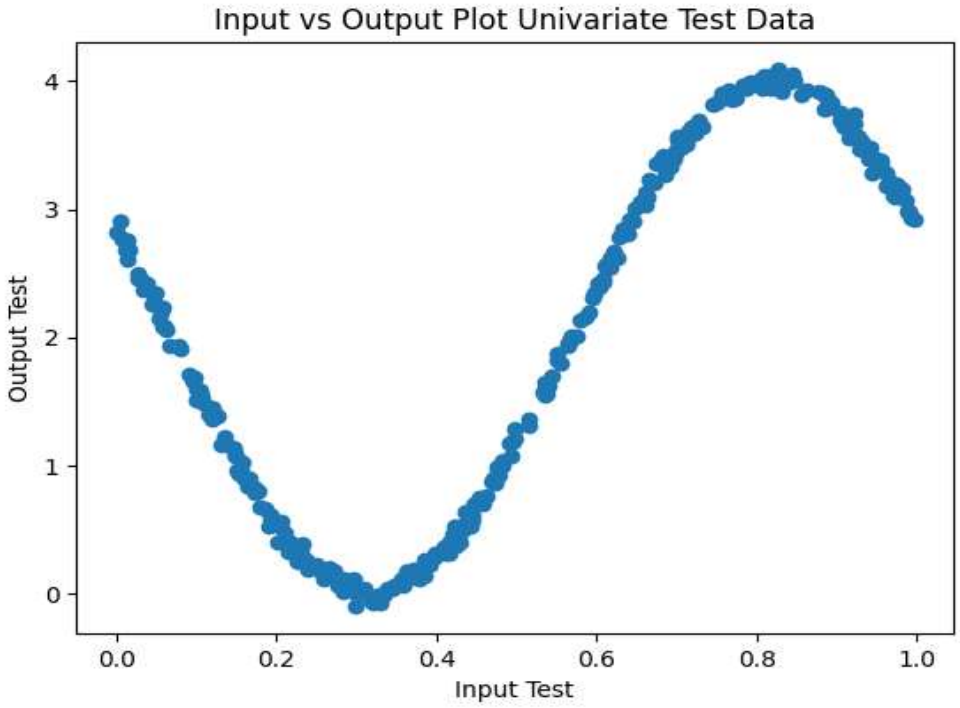
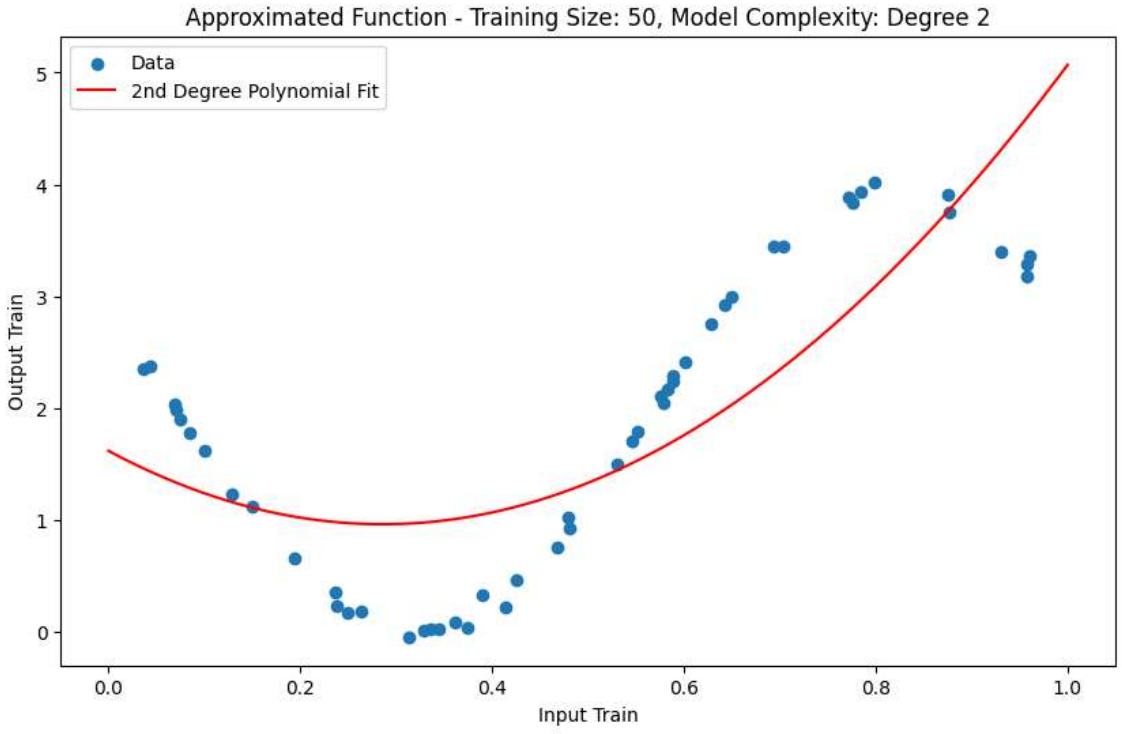


Fig59

- The values of x roughly lie between 0 and 1 (they being included).
- The y values roughly lie between -0.14576 and 4.1118.
- The number of test data points is 300 (whole 30% of 1001).

We can observe from the train and test data that a linear or quadratic polynomial won't be a good fit and we need at least a cubic polynomial or more for a good fit.

1. **For degree/model complexity being 2**



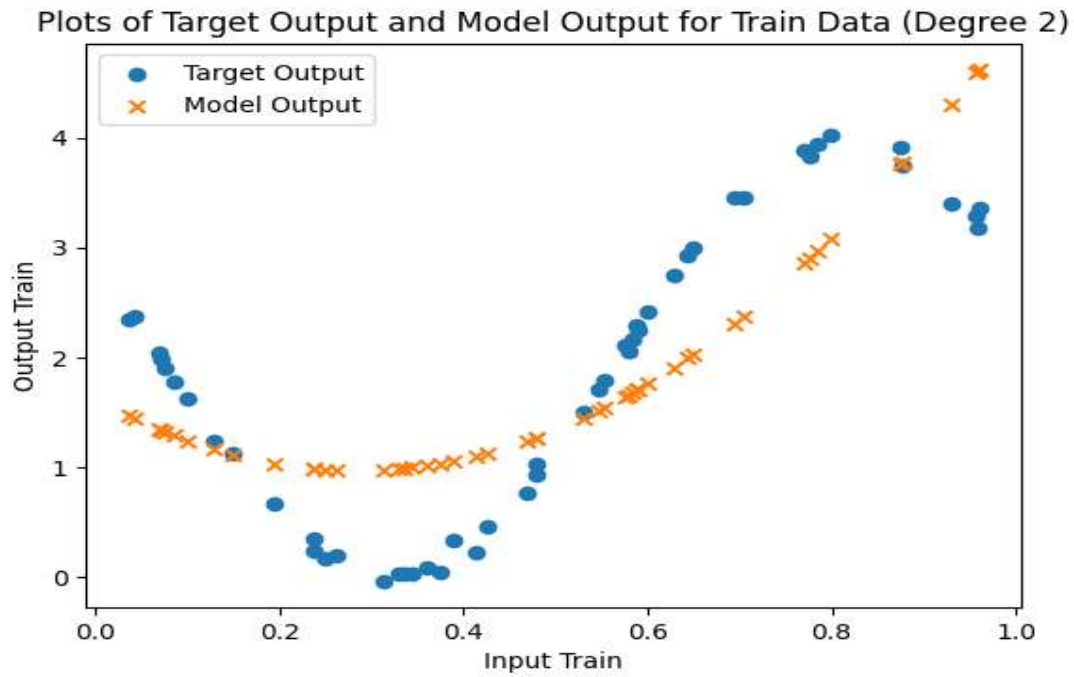


Fig60 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 2)

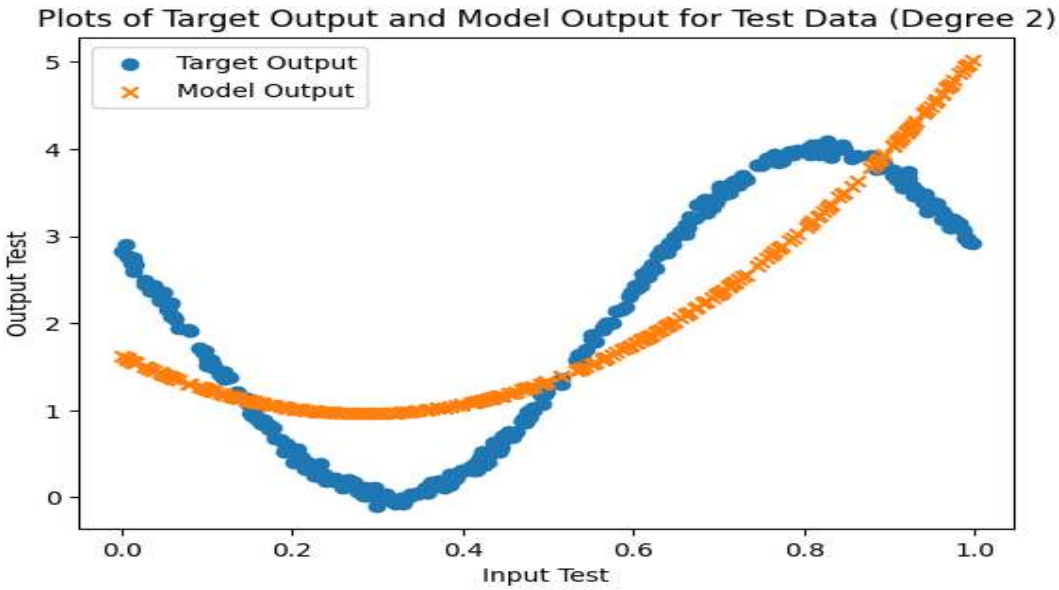


Fig61 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 2)

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 2:	1.619190943628821
w1 value of degree 2:	-4.592939340016411
w2 value of degree 2:	8.04314823569151
MSE for degree 2 training data:	0.601578192407018
MSE Percent for degree 2 training data:	32.58032782040147
MSE for degree 2 test data:	0.6664674828698822
MSE Percent for degree 2 test data:	33.321299029153984

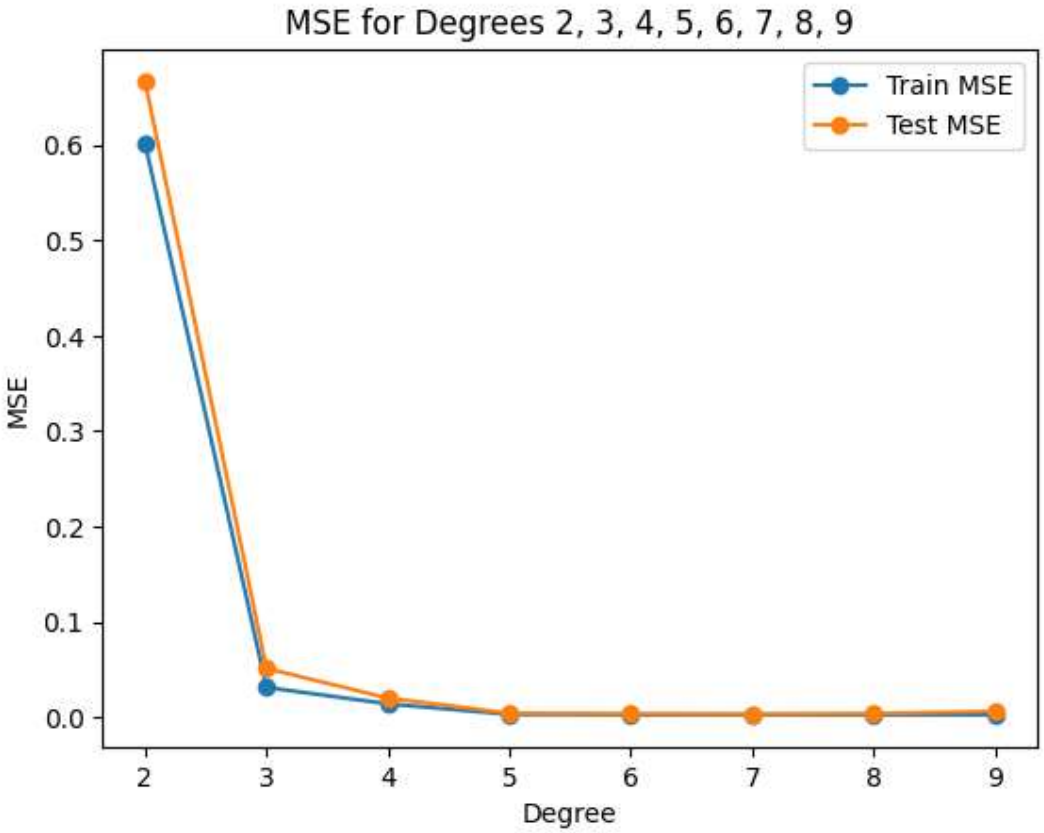
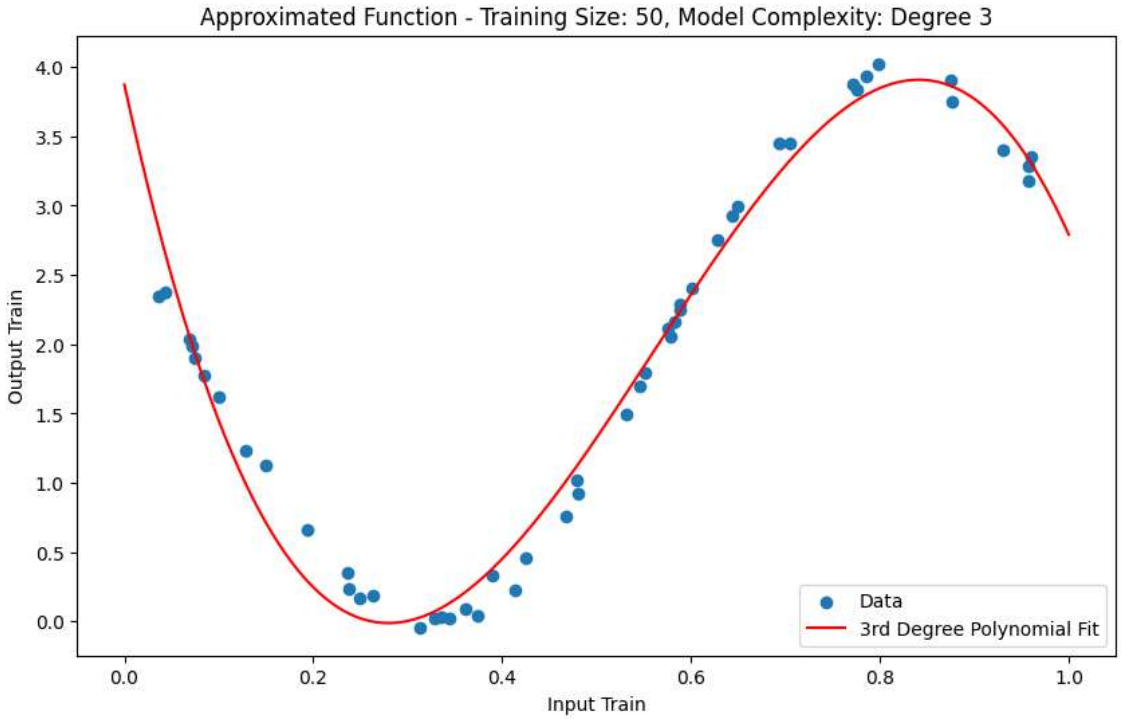


Fig62 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

2. **For degree/model complexity being 3**



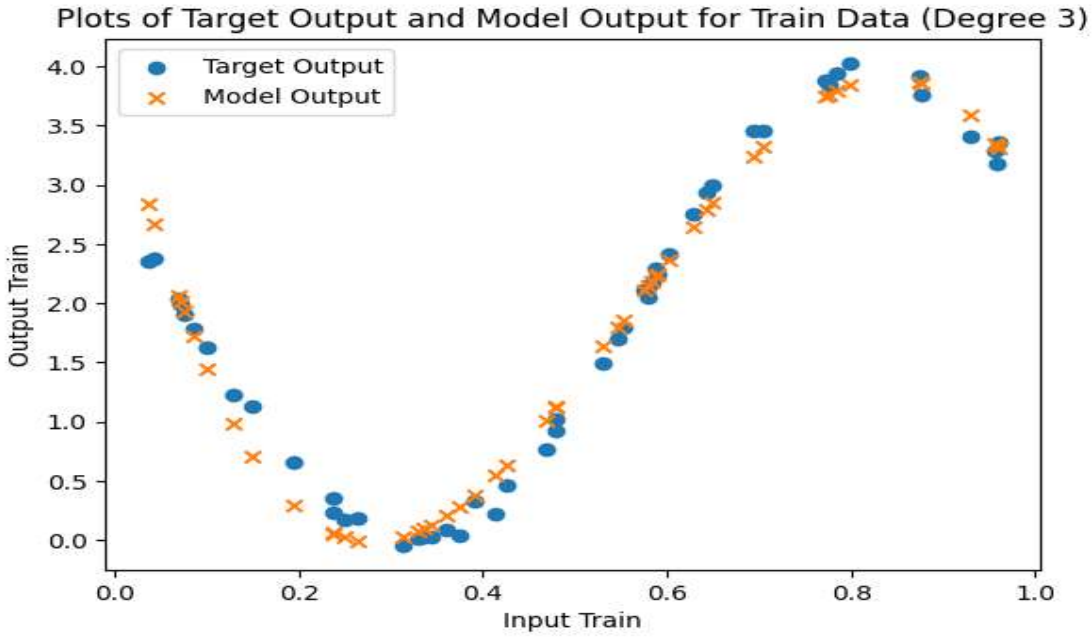


Fig63 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 3)

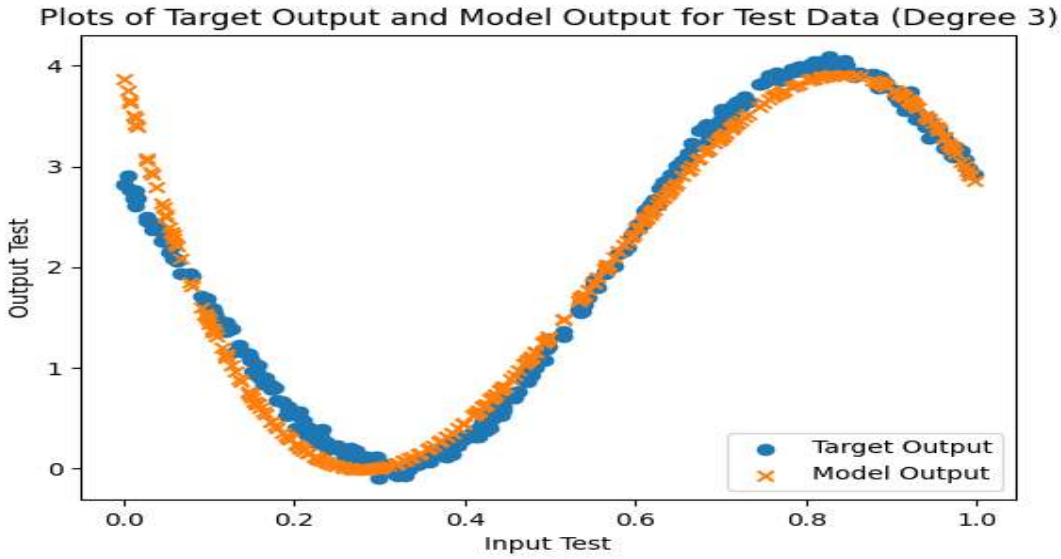


Fig64 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 3).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 3:	3.8732191941333105
w1 value of degree 3:	-31.248169594499654
w2 value of degree 3:	74.4396239710505
w3 value of degree 3:	-44.27321239063749
MSE for degree 3 training data:	0.03156720809559771
MSE Percent for degree 3 training data:	1.7096197985740123
MSE for degree 3 test data:	0.05148487192857463
MSE Percent for degree 3 test data:	2.574083293039917

We observe that **degree 3 is a better fit than degree 2** and **both train and test MSE have reduced significantly (steep decline)** for instance from **degree 2 to 3 train error has reduced from 0.601 to 0.0315** and **for test data test error has reduced from 0.666 to 0.0514.**

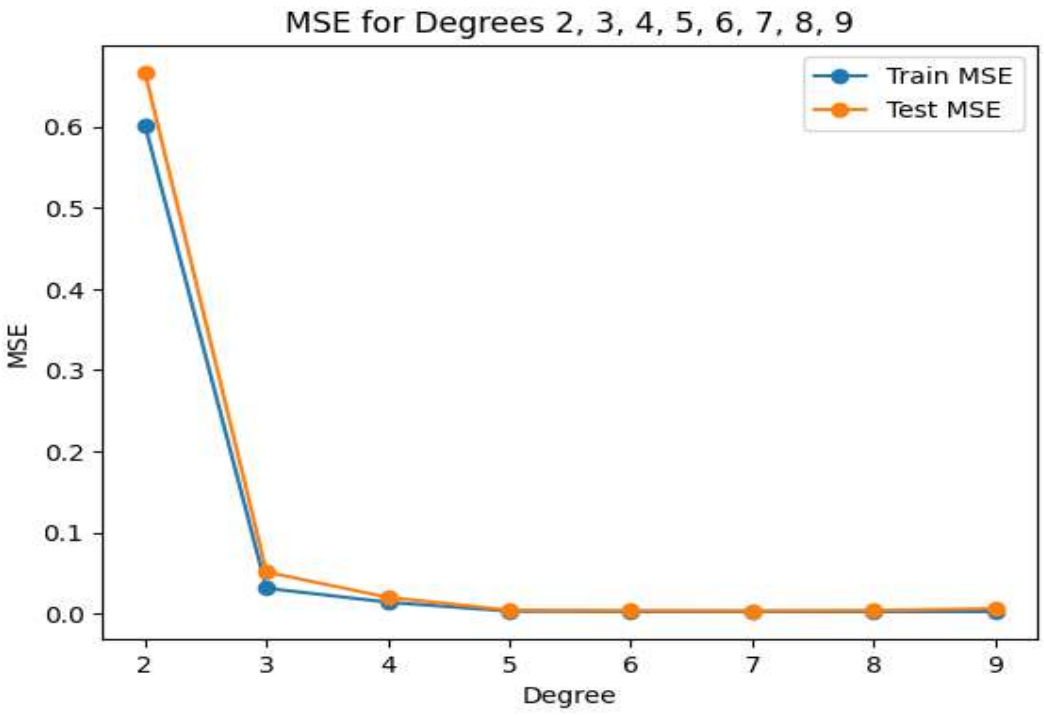
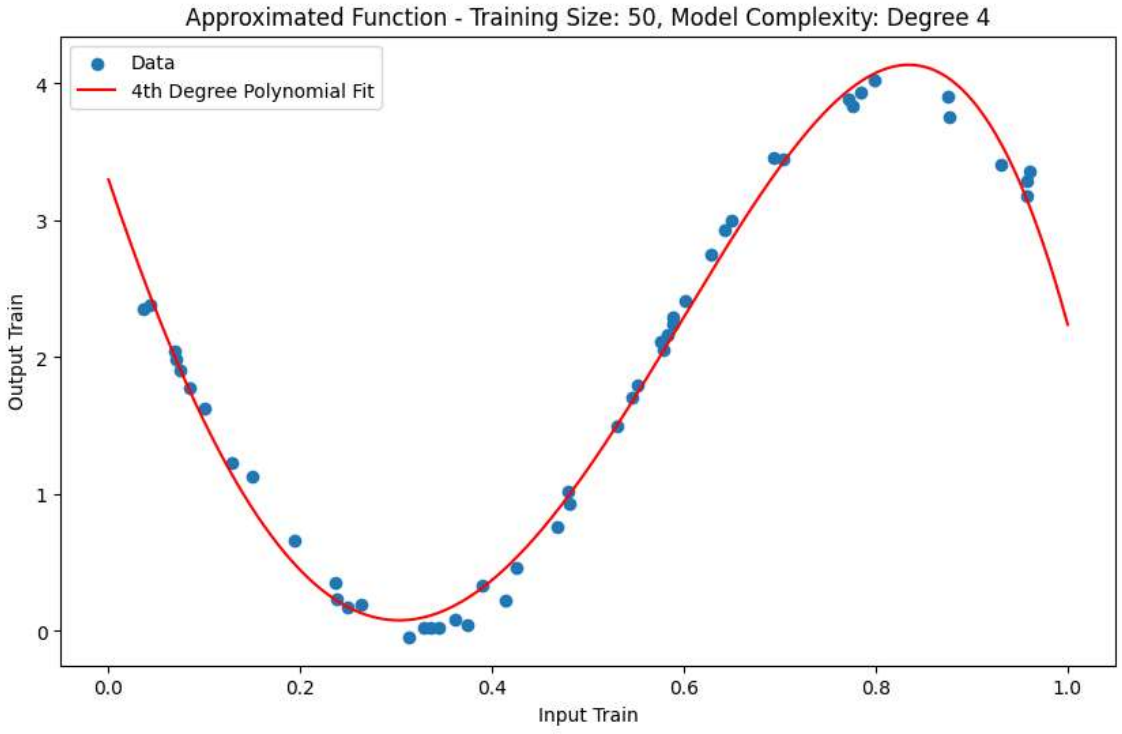


Fig65 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

3. For degree/model complexity being 4



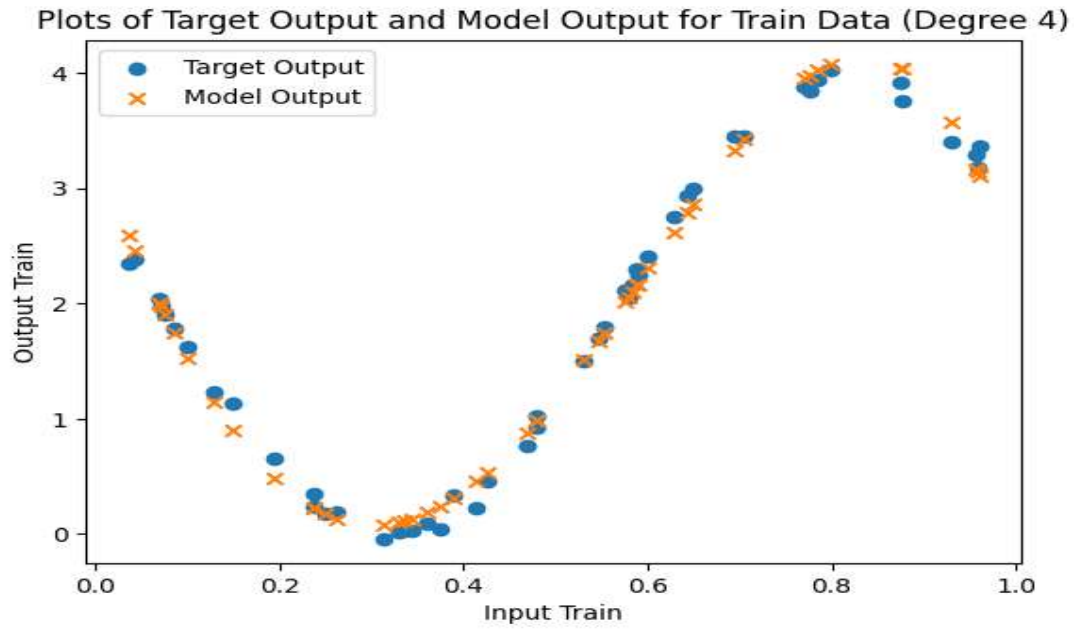


Fig66 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 4).

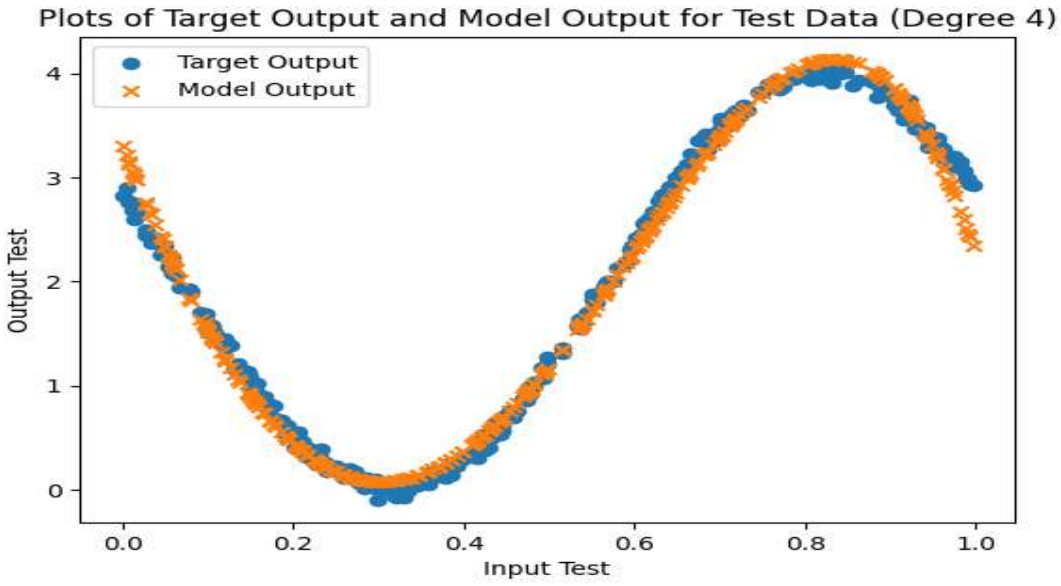


Fig67 **Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 4).**

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 4:	3.297822859329233
w1 value of degree 4:	-20.852782539788883
w2 value of degree 4:	29.188538351004947
w3 value of degree 4:	25.678107353690645
w4 value of degree 4:	-35.073100981209365
MSE for degree 4 training data:	0.014320454642178009
MSE Percent for degree 4 train data:	0.775568517390137
MSE for degree 4 test data:	0.020385103435811604
MSE Percent for degree 4 test data:	1.019191700696263

We observe that:-

- Degree 4 is a better fit than degree 3.
- Both train and test MSE have reduced from degree 3 to 4 but not as drastically as they did from degree 2 to 3.

Moving onto 4 degree from 3 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 2 to 3 which is the case as observed from the values of train and test MSE.

This indicates the model has probably become a decent fit for the data and further increase in degrees may not be that beneficial as it may lead to:

- **Overfitting (model may not be able to generalise).**
- **More complexity**.
- **Requirement of more training samples** (the thumb rule mentioned earlier).
- **Very less reduction in train and test MSE values**.
- **Huge coefficient values which in other words means incurring the overhead/cost of regularisation in case of the overfitting.**

Whether these things actually occur in the case of train samples being 50 (a sufficient amount for 5 degree polynomials but not for degrees above) is becoming evident as we are proceeding ahead and obtaining results for higher degrees.

A point to observe is that the values of our parameters (w0, w1, w2, w3) are not that extreme, hence apparently there is no need of regularisation as of now.

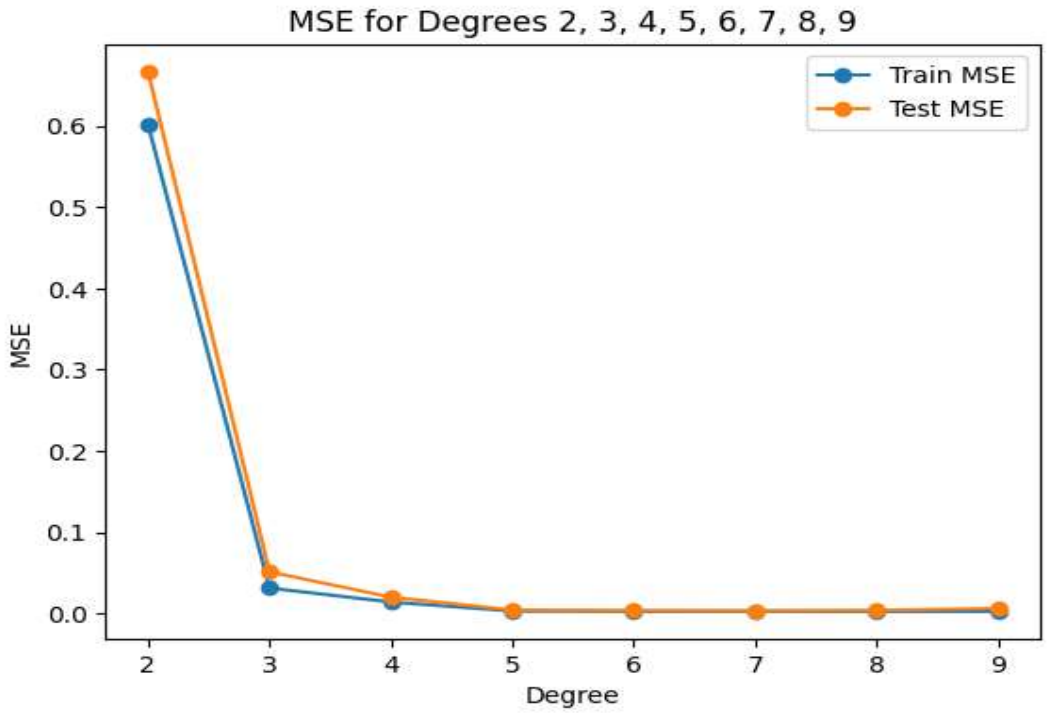


Fig68 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

4. **For degree/model complexity being 5**

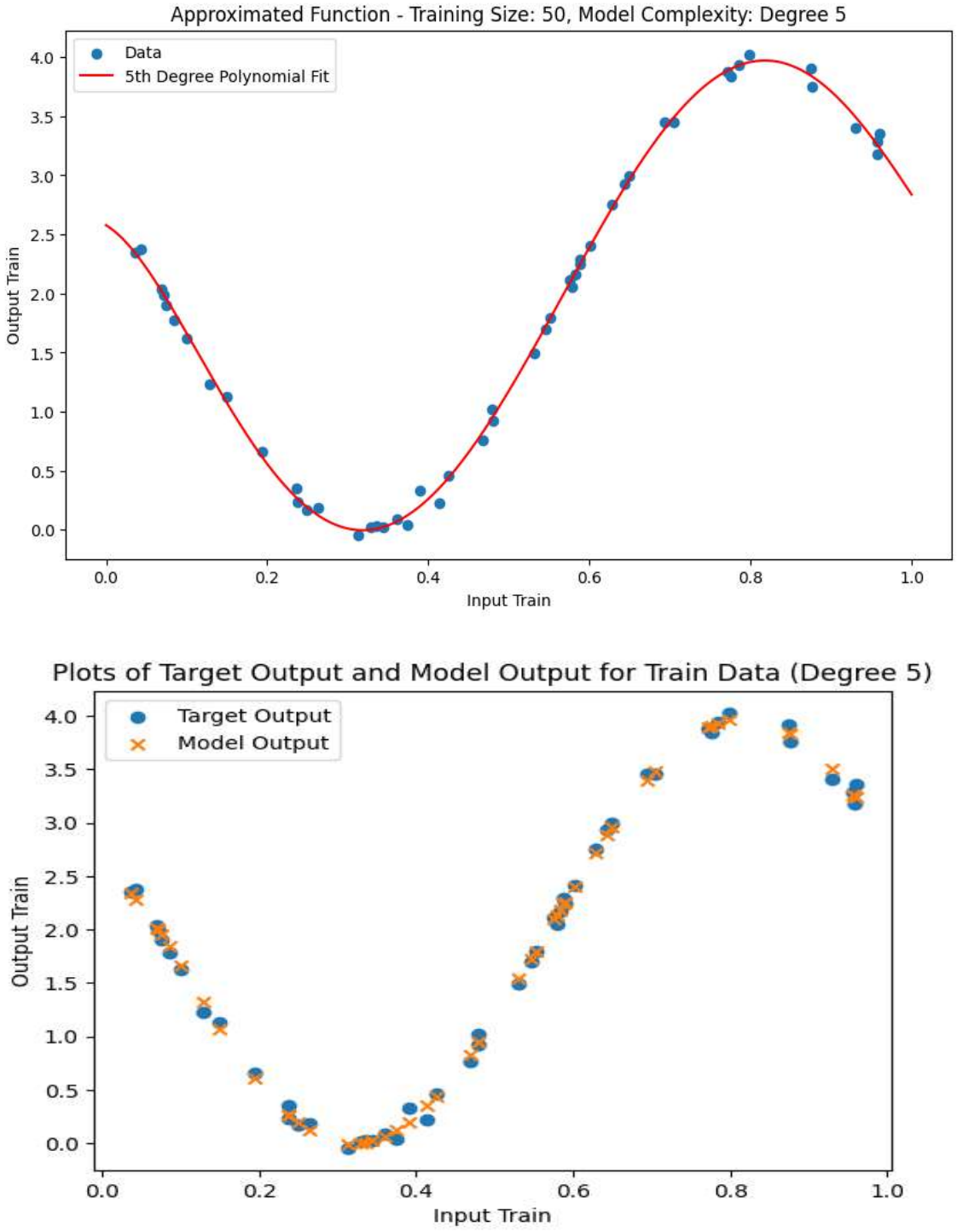


Fig69 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 5)

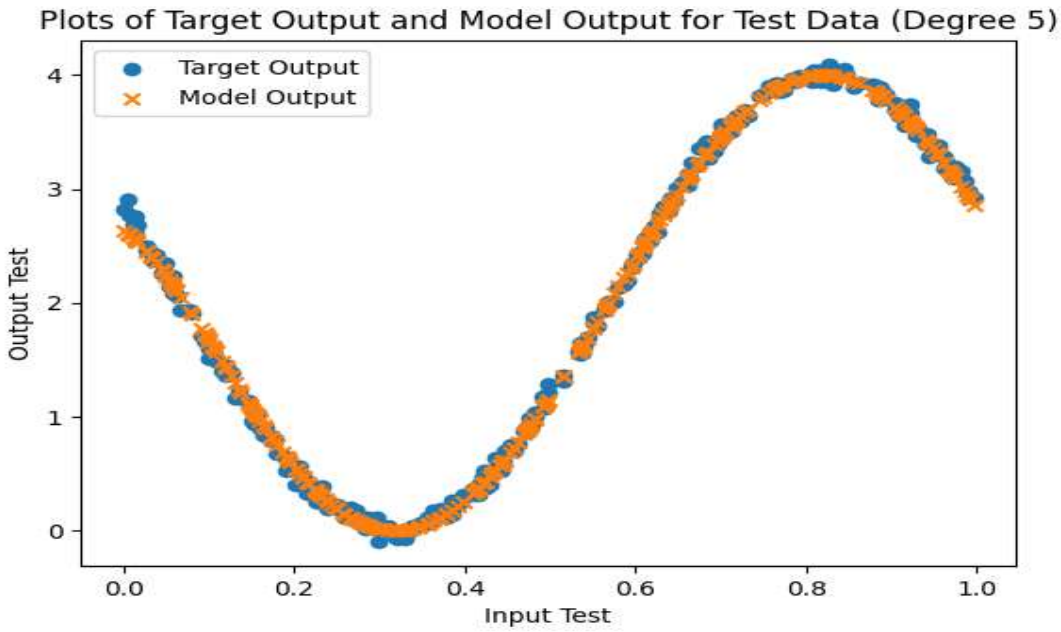


Fig70 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 5).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 5: 2.5775052882968157

w1 value of degree 5: -4.09074942230016

w2 value of degree 5: -76.74544046685736

w3 value of degree 5: 295.25375138501227

w4 value of degree 5: -332.3140608441816

w5 value of degree 5: 118.15624209245397

MSE for degree 5 training data: 0.0033512701588761236

MSE Percent for degree 5 train data: 0.18149840165256506

MSE for degree 5 test data: 0.004544554111921757

MSE Percent for degree 5 test data: 0.22721355566432144

We observe that:-

- Degree 5 is a better fit than degree 4.
- Both train and test MSE have reduced from degree 4 to 5 but not as drastically as they did from degree 3 to 4.

Moving onto 5 degree from 4 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 3 to 4 which is the case as observed from the values of train and test MSE.

This indicates the model has probably become a decent fit for the data and <u>further increase in degrees may not be that beneficial</u> as it may lead to:

- **Overfitting (model may not be able to generalise).**
- **More complexity**.
- **Requirement of more training samples** (the thumb rule mentioned earlier).
- **Very less reduction in train and test MSE values**.
- **Huge coefficient values which in other words means incurring the overhead/cost of regularisation in case of the overfitting.**

The values of our parameters have increased significantly because the number of training samples are limited.

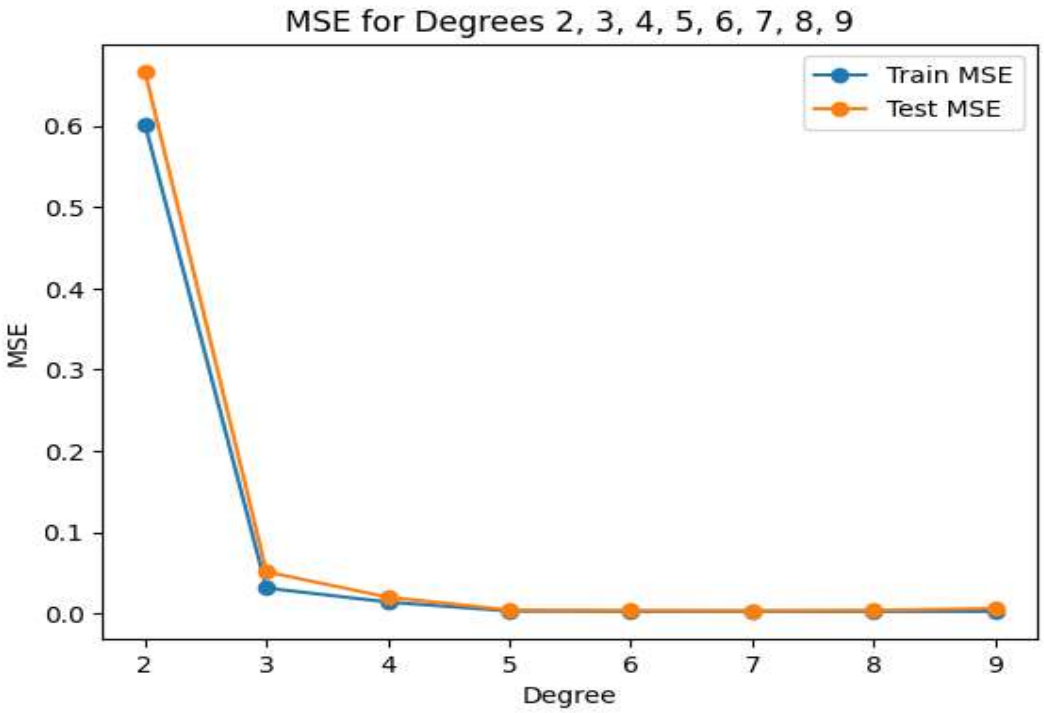


Fig71 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

5. **For degree/model complexity being 6**

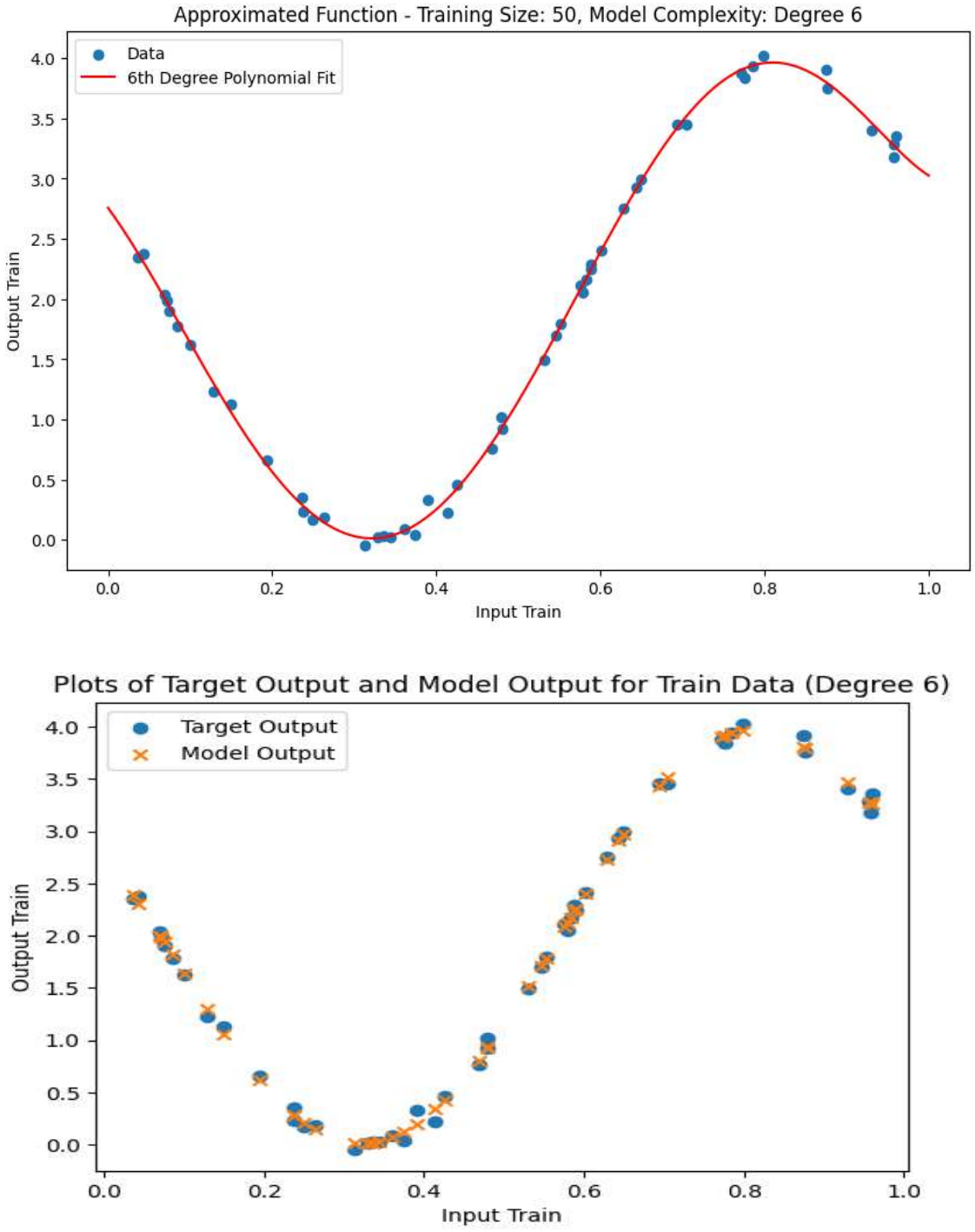


Fig72 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 6)

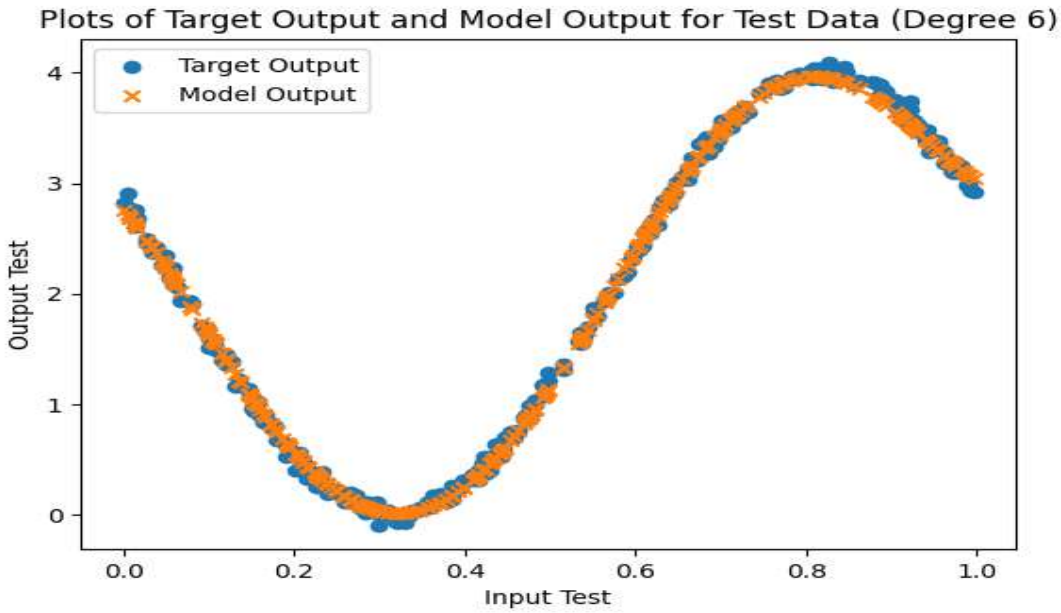


Fig73 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 6).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 6:	2.757600814595835
w1 value of degree 6:	-9.35848708847566
w2 value of degree 6:	-29.99359027129954
w3 value of degree 6:	116.13050625041889
w4 value of degree 6:	-0.19447369115363994
w5 value of degree 6:	-175.2853819395411
w6 value of degree 6:	98.96979275431674
MSE for degree 6 training data:	0.0029641517786962433
MSE Percent for degree 6 training data:	0.16053280833360056
MSE for degree 6 test data:	0.003949596870854018
MSE Percent for degree 6 test data:	0.19746754607085854

We observe that:-

- Degree 6 is a better fit than degree 5.
- Both train and test MSE have reduced from degree 5 to 6 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.

Moving onto 6 degree from 5 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 4 to 5 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have not increased significantly from degree 5 to 6** and the **error has declined but it is significantly less** which means **we are making our model more complex, it is losing its generalisation ability and the coefficients might soon be large at any point so we should select some previous degree model as a suitable one.**

That being said, our test MSE is still low since we have a decent number of training samples but the model is still losing its generalisation ability.

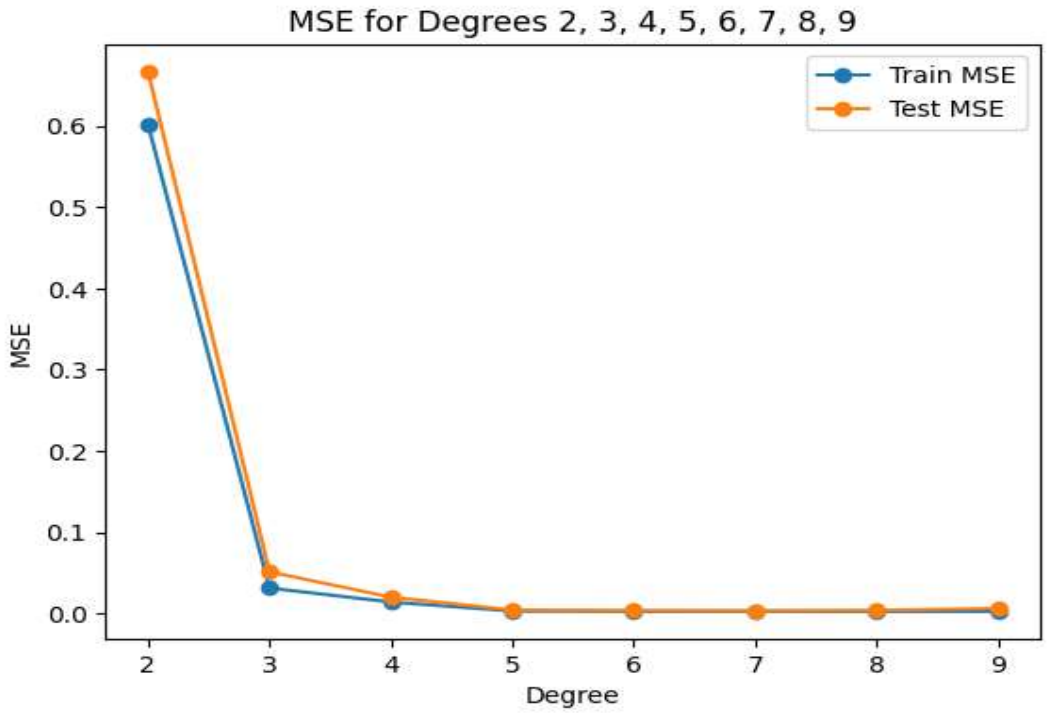


Fig74 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

6. **For degree/model complexity being 7**

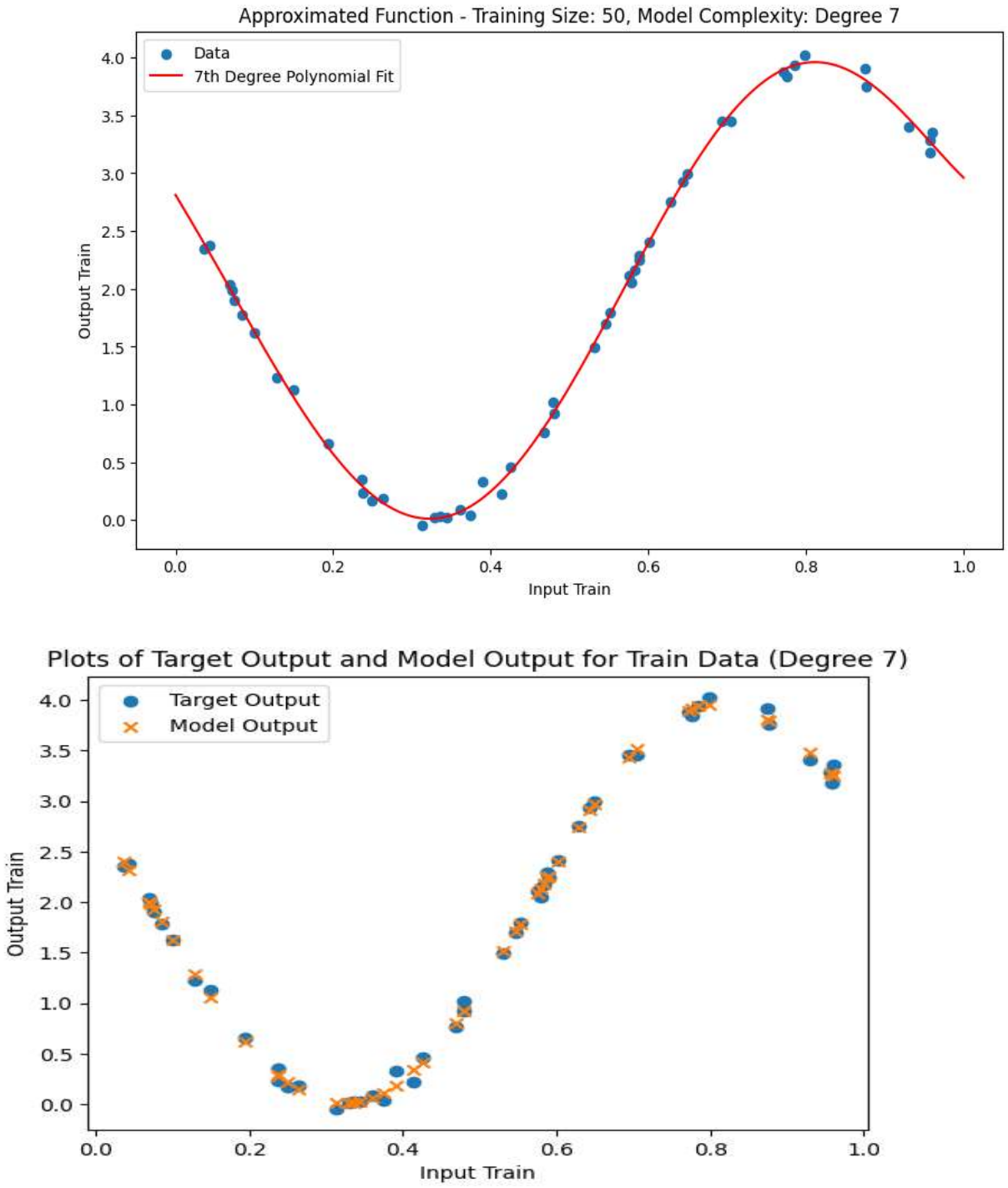


Fig75 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 7)

Fig76 **Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 7).**

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 7: 2.8123770135638653

w1 value of degree 7: -11.296074989098999

w2 value of degree 7: -7.719404882852341

w3 value of degree 7: 0.8165402607289913

w4 value of degree 7: 307.28110861074066

w5 value of degree 7: -613.8746719080343

w6 value of degree 7: 416.74901762401646

w7 value of degree 7: -91.8076881294906

MSE for degree 7 training data: 0.00294323743208453

MSE Percent for degree 7 train data: 0.1594001272002757

MSE for degree 7 test data: 0.0036461347849865515

MSE Percent for degree 7 test data: 0.18229538663757555

We observe that:-

- Degree 7 is a better fit than degree 6.
- Both train and test MSE have reduced from degree 6 to 7 but not as drastically as they did from degree 5 to 6 and the reduction is extremely minute.
- Degree 6 to 7 train error has almost stayed the same.
- For test data, test error has almost stayed the same.

Moving onto 7 degree from 6 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 5 to 6 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased from degree 6 to 7** and the **MSE values have stayed the same so we should have stopped earlier/ at degree 5 or 4 as discussed earlier (since degree 5 seems to have less generalisation capacity as compared to degree 4).**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and train MSE and the model most probably has now lost its generalisation ability.

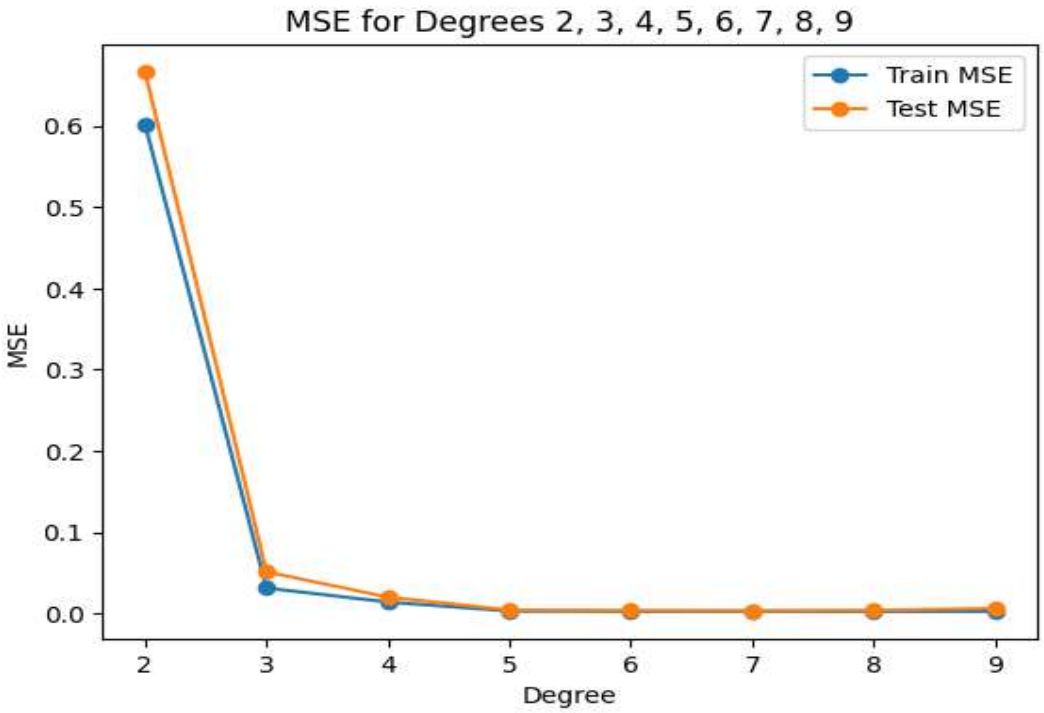


Fig77 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing complexity of the model which may result in huge values of parameters, but since we have just sufficient number of training samples so in theory we should be able to observe no increase in test MSE.

And as we see that is the case for results of degrees we see ahead.

7. **For degree/model complexity being 8**

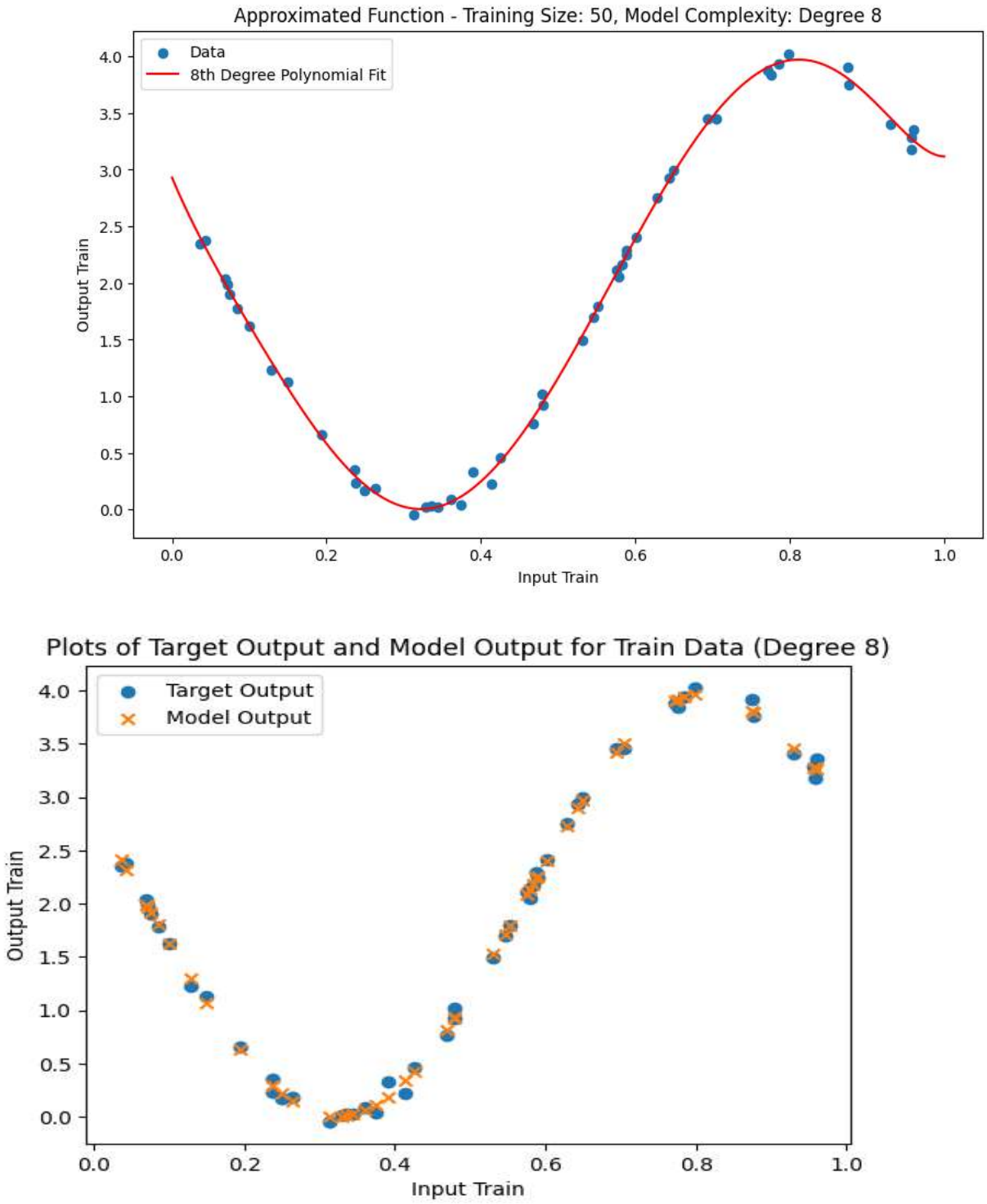


Fig78 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 8)

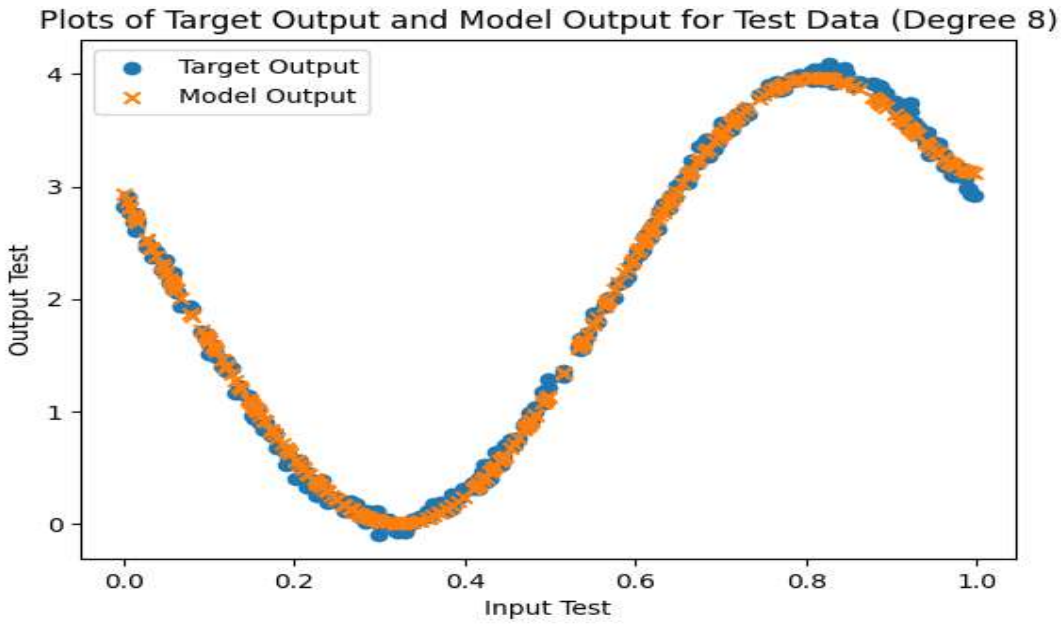


Fig79 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 8).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 8: 2.9311131862060296

w1 value of degree 8: -16.348036031398724

w2 value of degree 8: 65.47716548066273

w3 value of degree 8: -492.0138694496357

w4 value of degree 8: 2082.4256203780087

w5 value of degree 8: -4244.919463574832

w6 value of degree 8: 4640.1270708141565

w7 value of degree 8: -2691.8780859342023

w8 value of degree 8: 657.3155036756509

MSE for degree 8 training data: 0.002896003796852824

MSE Percent for degree 8 train data: 0.15684204358052062

MSE for degree 8 test data: 0.004182354021742784

MSE Percent for degree 8 test data: 0.20910467890221784

We observe that:-

- Degree 8 is a better fit than degree 7.
- Both train and test MSE have reduced from degree 7 to 8 but not as drastically as they did from degree 6 to 7 and the reduction is extremely minute.

Moving onto 8 degree from 7 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 6 to 7 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 7 to 8** and the **MSE values have stayed the same.**

We are now just increasing our model complexity and values of our parameters. In fact we are not having any gains in terms of test and training MSE and the model most probably has now completely lost its generalisation ability.

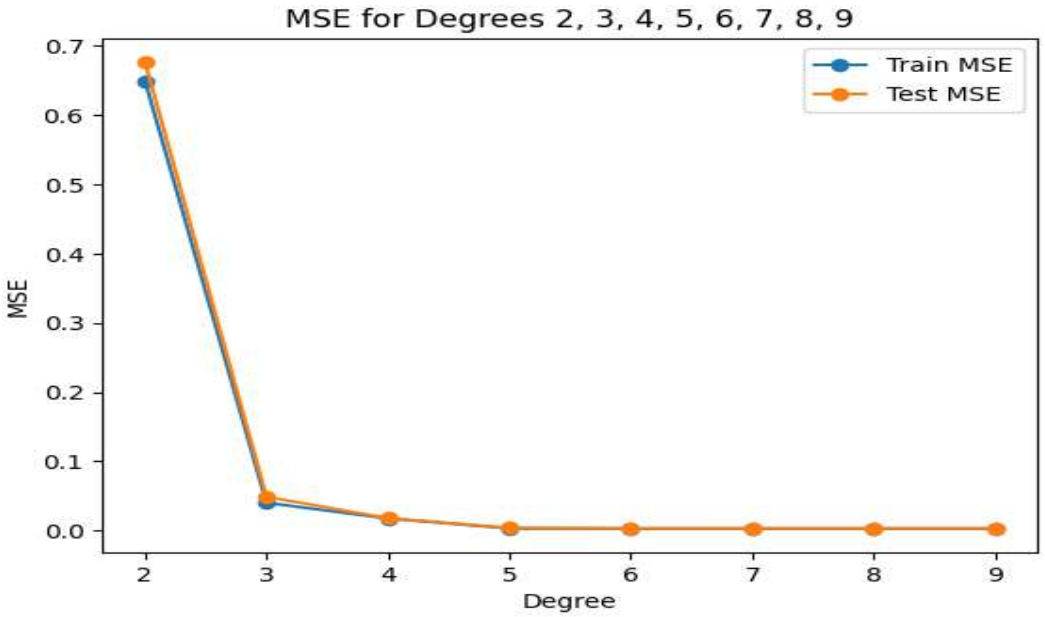
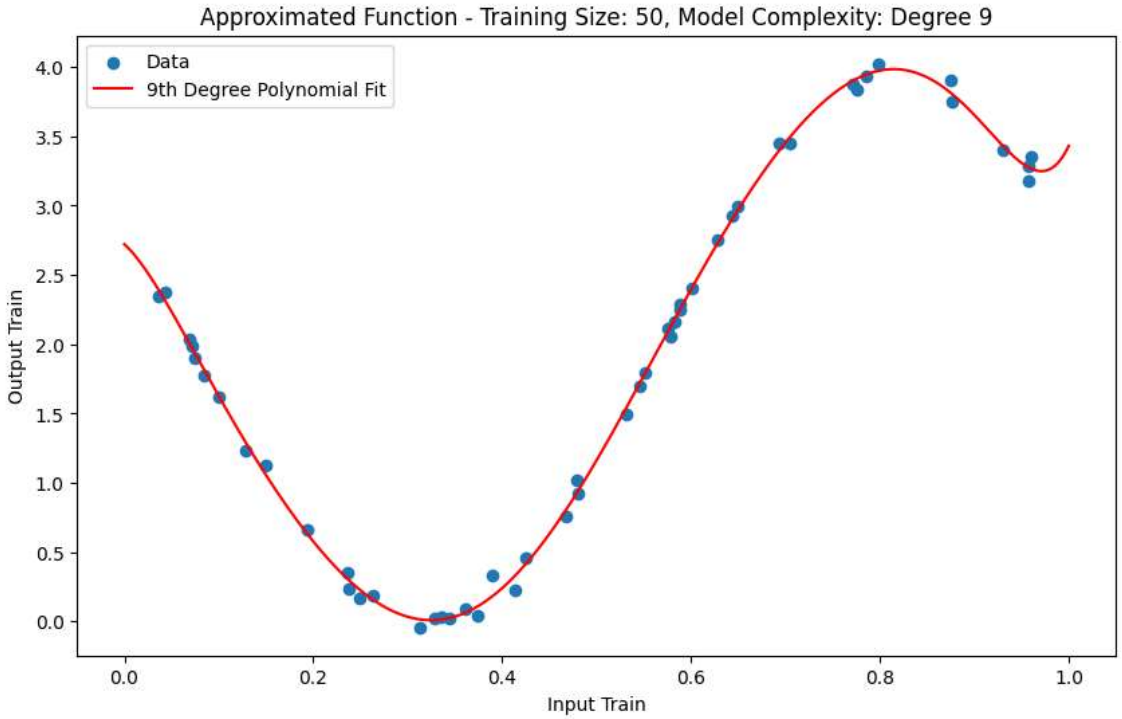


Fig80 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which is resulting in huge values of parameters.

8. **For degree/model complexity being 9**



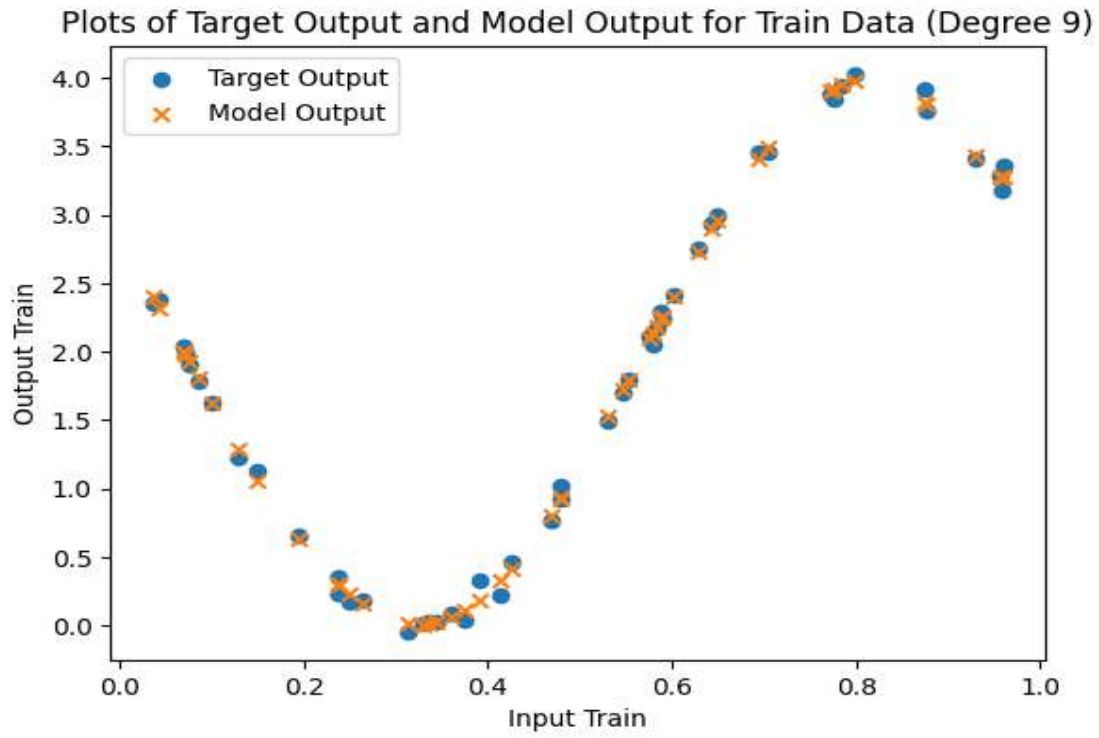


Fig81 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 9)

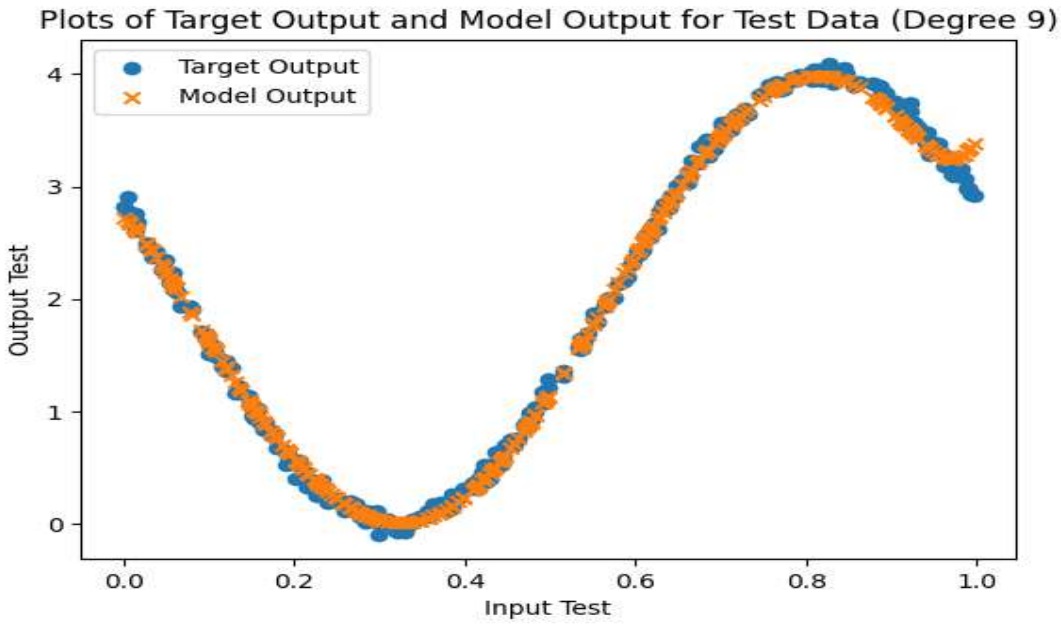


Fig82 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 9).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 9:	2.7218953711500915
w1 value of degree 9:	-6.1734073271791
w2 value of degree 9:	-108.91618783471723
w3 value of degree 9:	944.1550579502353
w4 value of degree 9:	-4466.636320774385
w5 value of degree 9:	13475.322674104467
w6 value of degree 9:	-24476.90680749584
w7 value of degree 9:	25798.176381291945
w8 value of degree 9:	-14594.815405698611
w9 value of degree 9:	3436.50389859967
MSE for degree 9 training data:	0.0028330987017302477
MSE Percent for degree 9 training data:	0.153435223575183
MSE for degree 9 test data:	0.006477518323224853
MSE Percent for degree 9 test data:	0.3238557477486697

We observe that:-

- Degree 9 is a better fit than degree 8.
- Both train and test MSE have reduced from degree 8 to 9 but not as drastically as they did from degree 7 to 8 and the reduction is extremely minute.
- Degree 8 to 9 train error has stayed the same.
- For test data, test error has stayed the same.

Moving onto 9 degree from 8 we expected a little decline in train and test MSE values as compared to the one we obtained from degree 7 to 8 which is the case as observed from the values of train and test MSE.

What we observe is that the values **of our parameters have increased significantly from degree 8 to 9** and the **MSE values have stayed the same.**

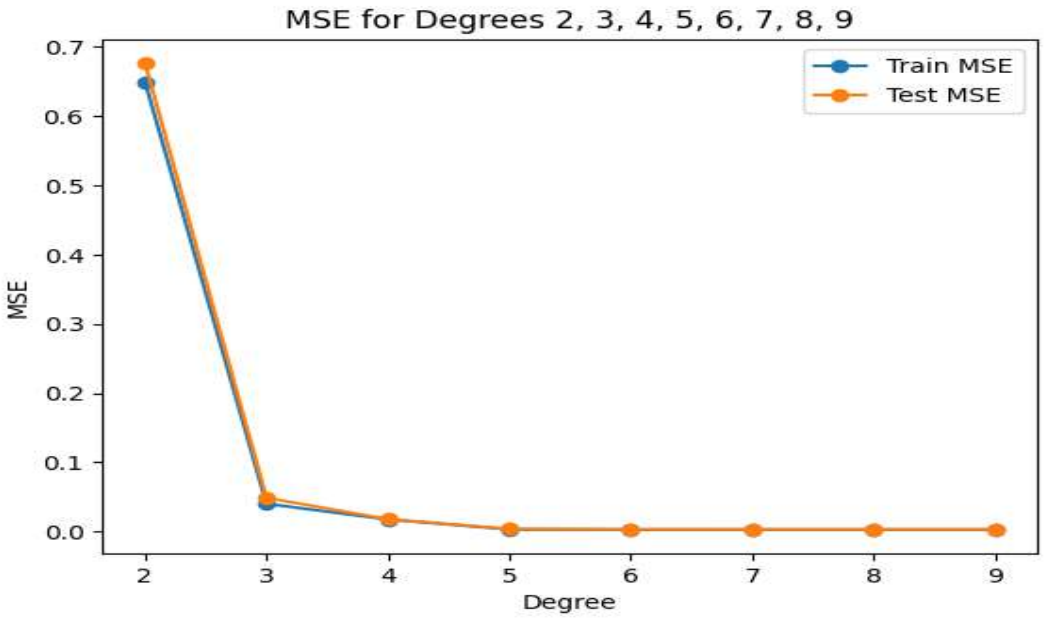
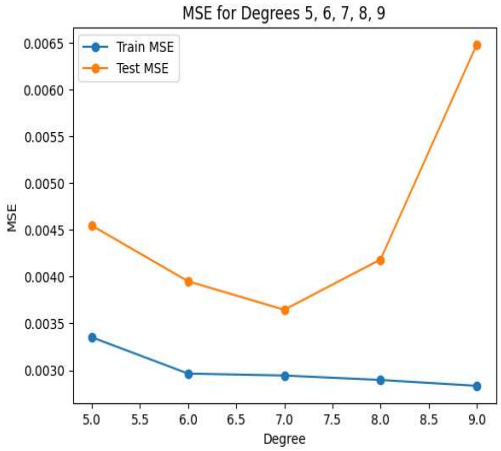


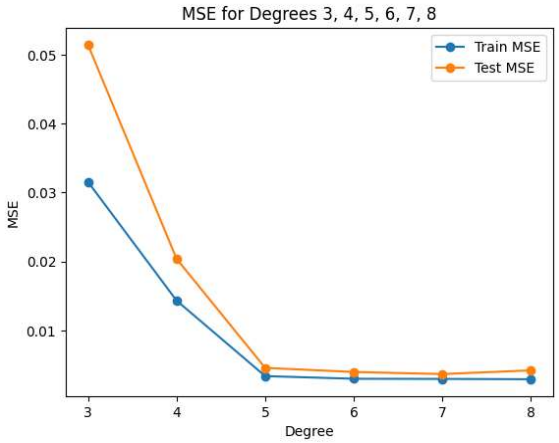
Fig83 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

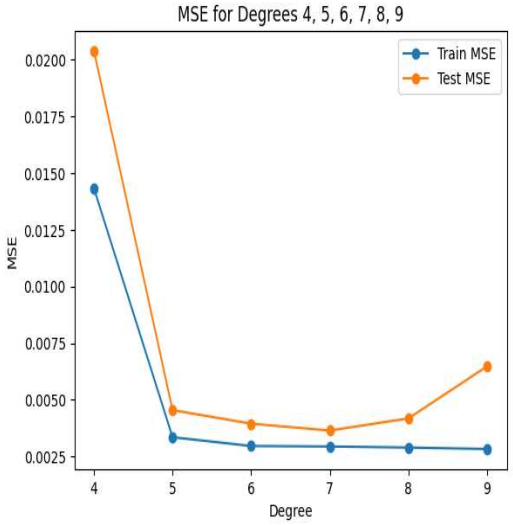
We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which is resulting in huge values of parameters.

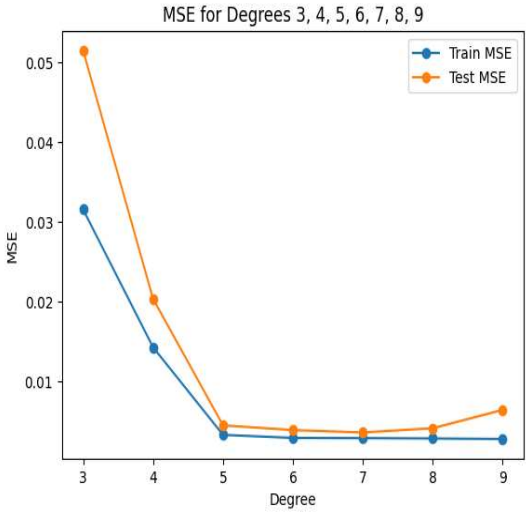
Regularization plots for train sample=50 and test samples=300

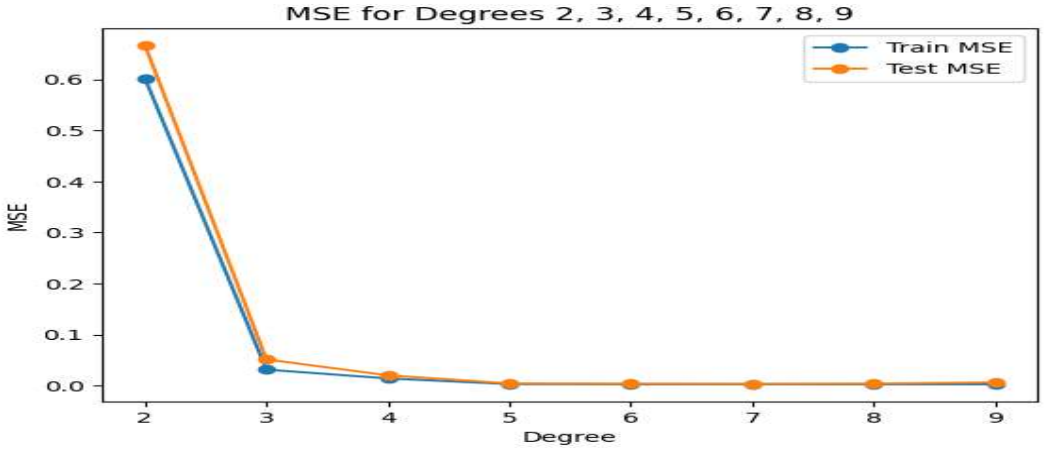
We can clearly observe all the necessary plots (where we are able to observe the difference) of MSE together.









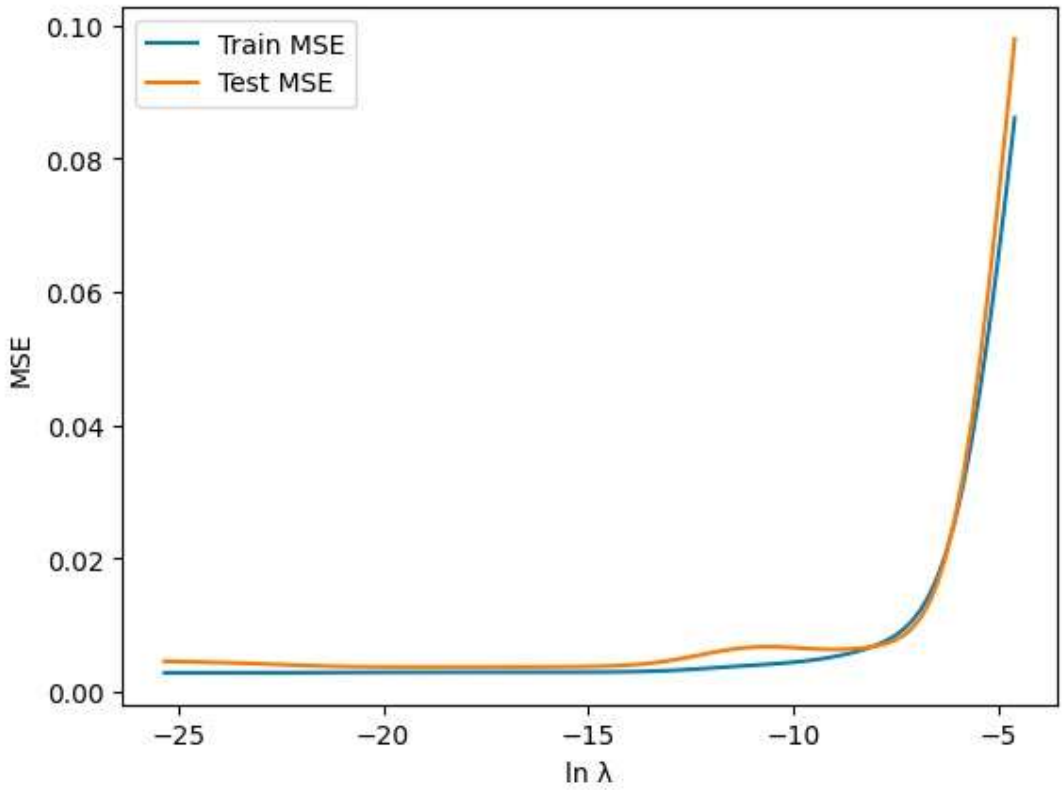


Regularization for degree 9 when number of train samples is 50 and test samples is 300

Now we select values of that lambda where test MSE is the lowest.

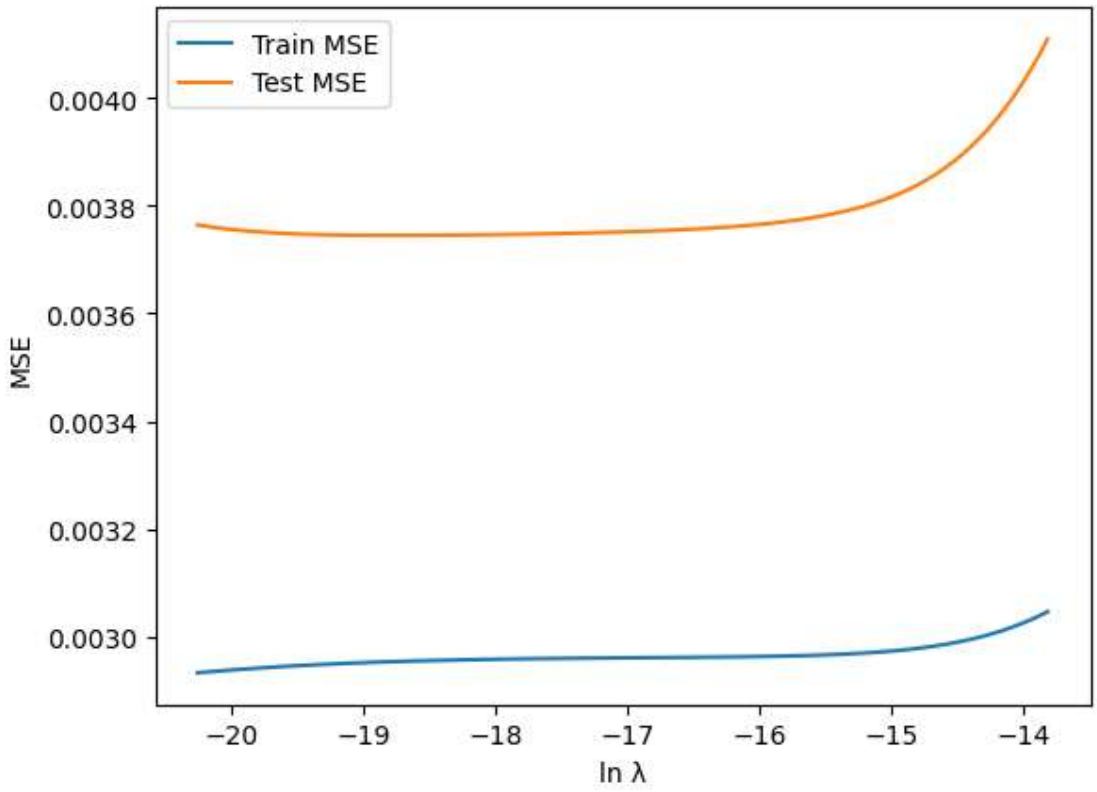
```
print(min(mse_train))
print(min(mse_test))
print(np.argmin(mse_test))
print(np.argmin(mse_train))
print(alpha_values[23])
print(alpha_values[0])
```

0.0029335738626301354
0.0037448432554714373
23
0
7.087695488385949e-09
1.584893192461111e-09



We run a code to find that value of lambda where our test MSE is minimum.

Here is that particular graph which has helped us find the global minima of test MSE for graph of ln lambda VS train and test MSE.



Essentially both train and test MSE for a certain value of lambda while regularising is lower than that of train and test MSE values without regularisation.

After regularization we obtain the following plot for **lambda=** 7.087695488385949e-09

Before Regularisation
w0 value of degree 9: 2.7218953711500915
w1 value of degree 9: -6.1734073271791
w2 value of degree 9: -108.91618783471723
w3 value of degree 9: 944.1550579502353
w4 value of degree 9: -4466.636320774385
w5 value of degree 9: 13475.322674104467
w6 value of degree 9: -24476.90680749584
w7 value of degree 9: 25798.176381291945
w8 value of degree 9: -14594.815405698611
w9 value of degree 9: 3436.50389859967

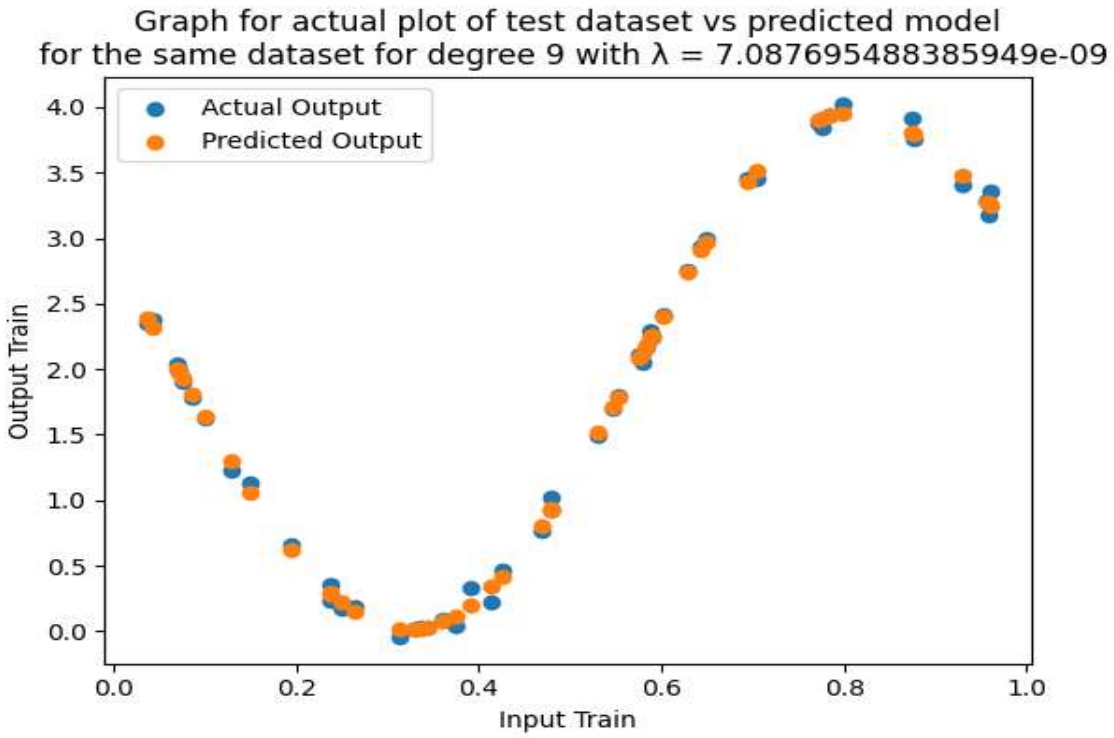
Before Regularisation:
MSE for degree 9 training data: 0.0028330987017302477
MSE Percent for degree 9 training data: 0.153435223575183
MSE for degree 9 test data: 0.006477518323224853
MSE Percent for degree 9 test data: 0.32385574774866

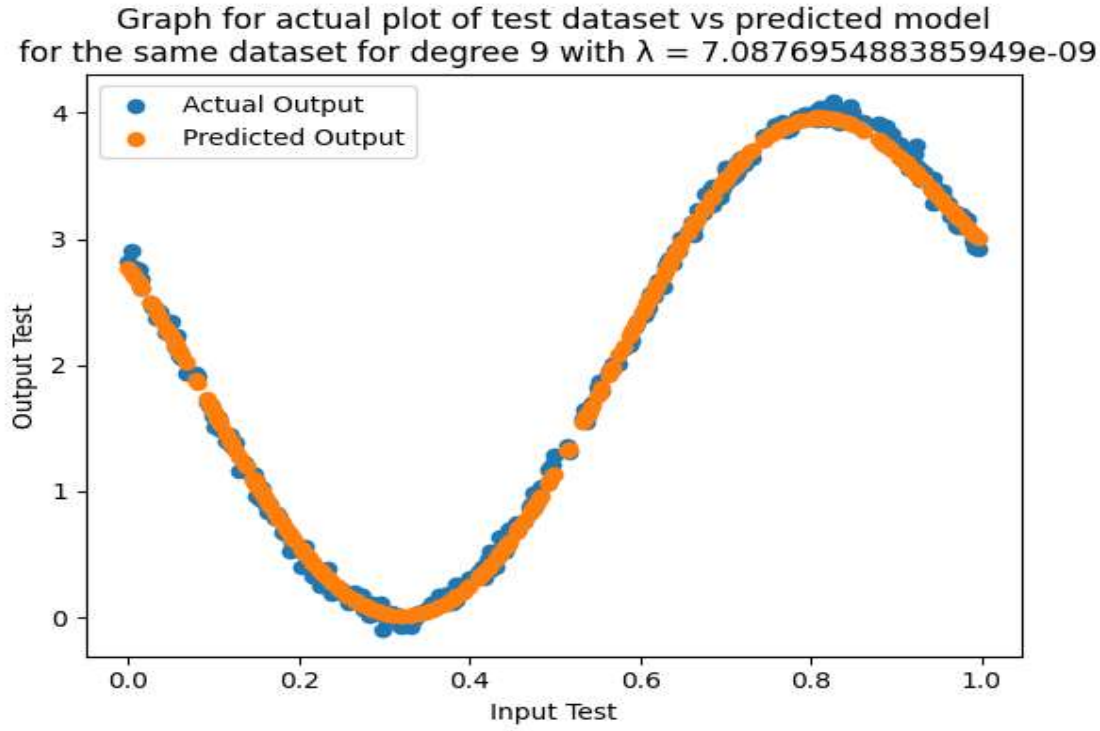
After regularization

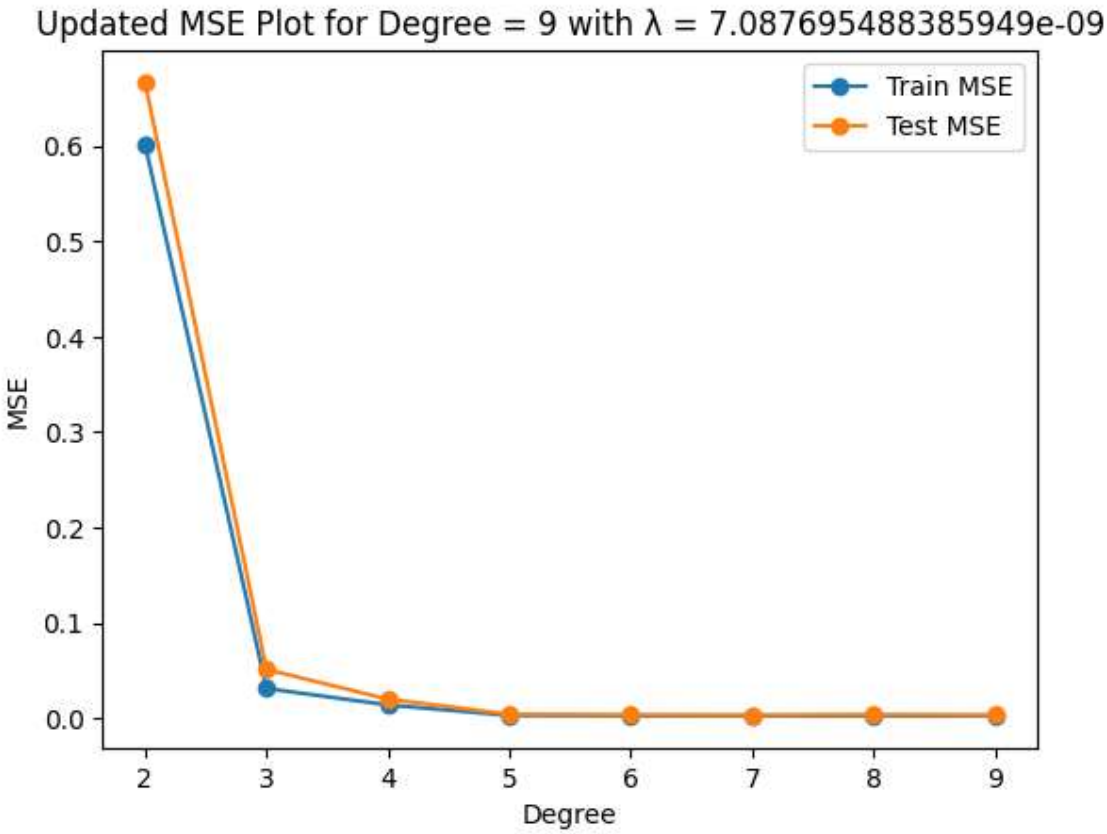
w0 value of degree 9 2.775877234776478
w1 value of degree 9 -9.97008482353357
w2 value of degree 9 -23.681522786423784
w3 value of degree 9 89.57004029908673
w4 value of degree 9 44.103451813697255
w5 value of degree 9 -172.98118668638475
w6 value of degree 9 1.3408535127078902
w7 value of degree 9 110.29874811652026
w8 value of degree 9 -37.1536999742994
w9 value of degree 9 -1.3246258824493422

After Regularisation:

MSE for degree 9 training data: 0.0029543547929466946
MSE Percent for degree 9 training data: 0.16000222226615193
MSE for degree 9 test data: 0.0037448432554714373
MSE Percent for degree 9 test data: 0.1872305028229506


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 7.087695488385949e-09
Actual Output
Predicted Output
Output Train
Input Train
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0.0
0.2
0.4
0.6
0.8
1.0


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 7.087695488385949e-09
Actual Output
Predicted Output
Output Test
Input Test
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0



Optimum graphs for **lambda=7.087695488385949e-09**

When number of train samples is 50 and test samples is 300.

Before Regularisation:

MSE for degree 9 training data: 0.0028330987017302477
MSE Percent for degree 9 training data: 0.153435223575183
MSE for degree 9 test data: 0.006477518323224853
MSE Percent for degree 9 test data: 0.32385574774866

After Regularisation:

MSE for degree 9 training data: 0.0029543547929466946
MSE Percent for degree 9 training data: 0.16000222226615193
MSE for degree 9 test data: 0.0037448432554714373
MSE Percent for degree 9 test data: 0.187230502822950

Note: When we consider 10 train samples.

Observations for 10 samples:

Plot for train data which is 10 train samples.

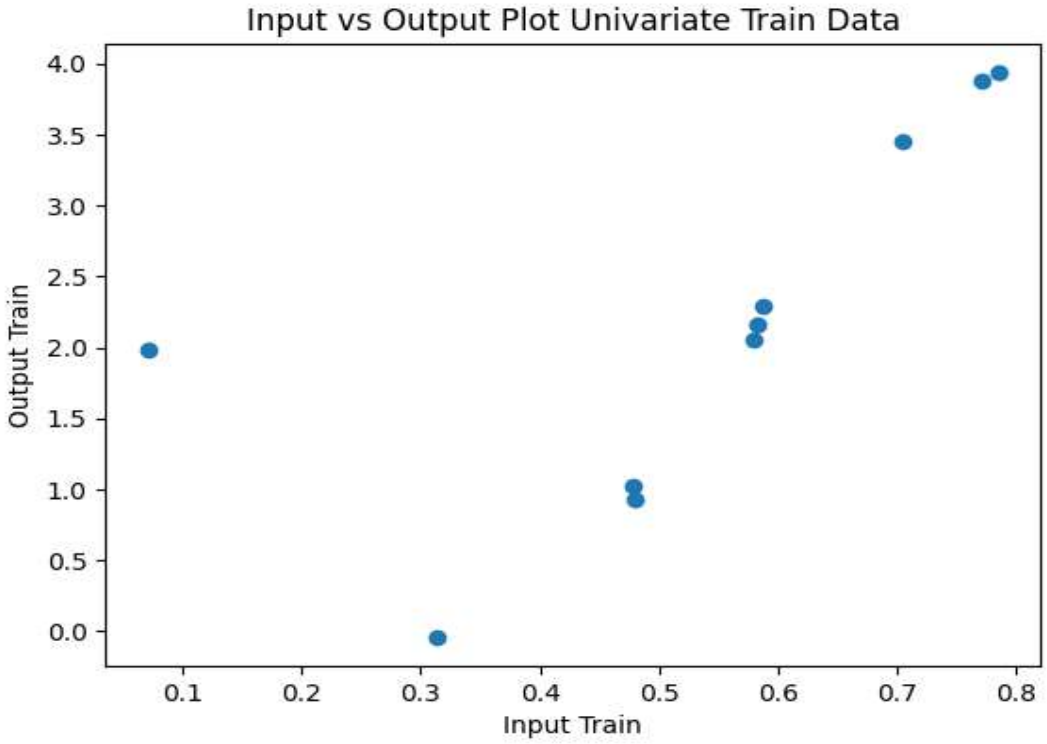


Fig84

- The values of x are between 0.05 and 0.78 (they being included).
- The y values should roughly lie between -0.01 and 3.9 The number of train data points is 10.
- The number of test samples is 300.

Plot of the test data (300 samples) for 10 train samples.

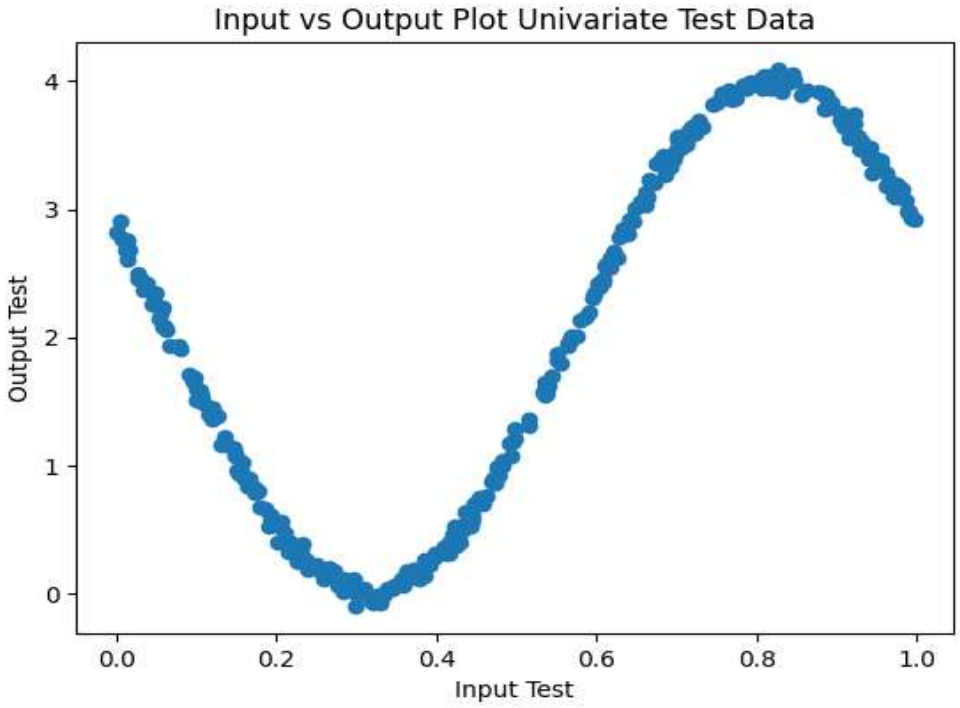
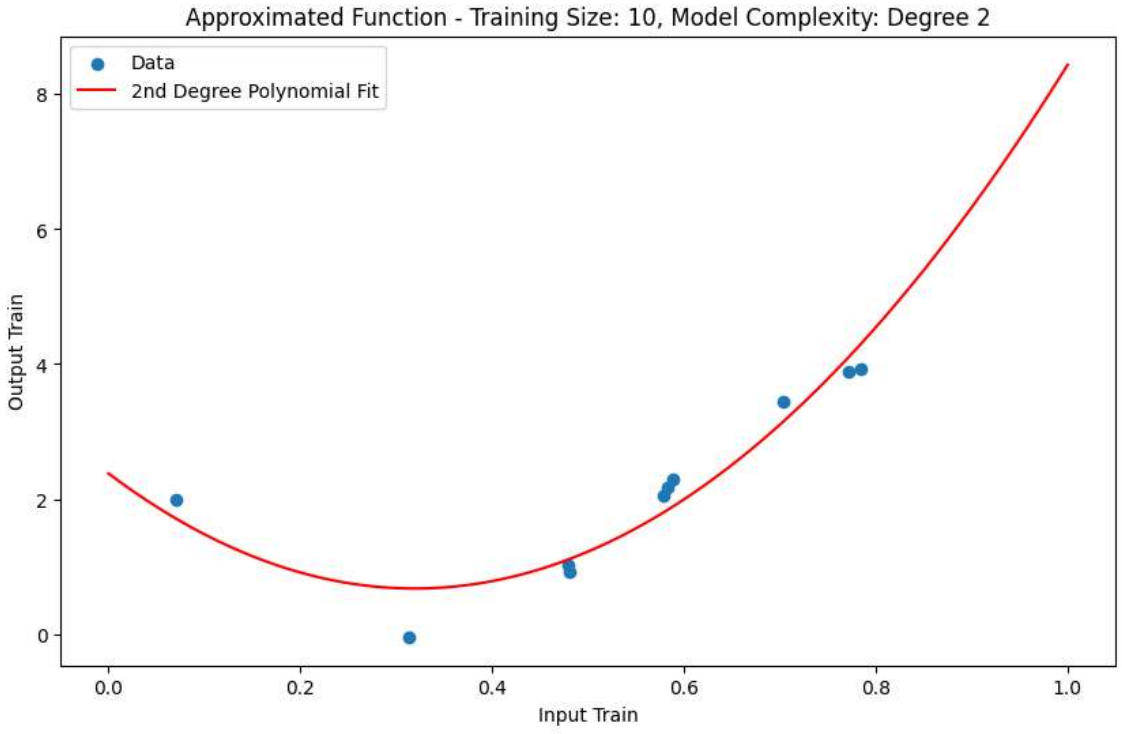


Fig85

- The values of x roughly lie between 0 and 1 (they being included).
- The y values roughly lie between -0.14576 and 4.1118.
- The number of test data points is 300 (whole 30% of 1001).

We can observe from the train and test data that a linear or quadratic polynomial won't be a good fit and we need at least a cubic polynomial or more for a good fit.

1. **For degree/model complexity being 2**



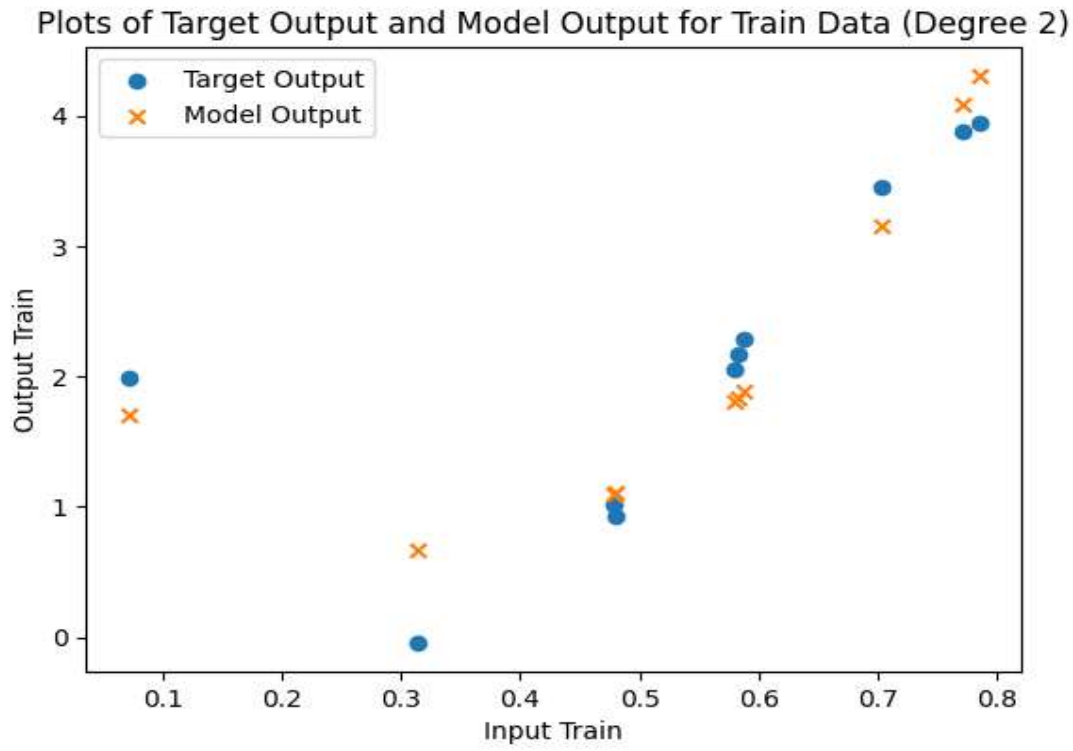


Fig86 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 2).

Fig87 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 2).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 2:	2.3789185751181248
w1 value of degree 2:	-10.68895814048925
w2 value of degree 2:	16.747128858649145
MSE for degree 2 training data:	0.12349109498783337
MSE Percent for degree 2 training data:	5.698879532774168
MSE for degree 2 test data:	2.1913093909883967
MSE Percent for degree 2 test data:	109.55864668460231

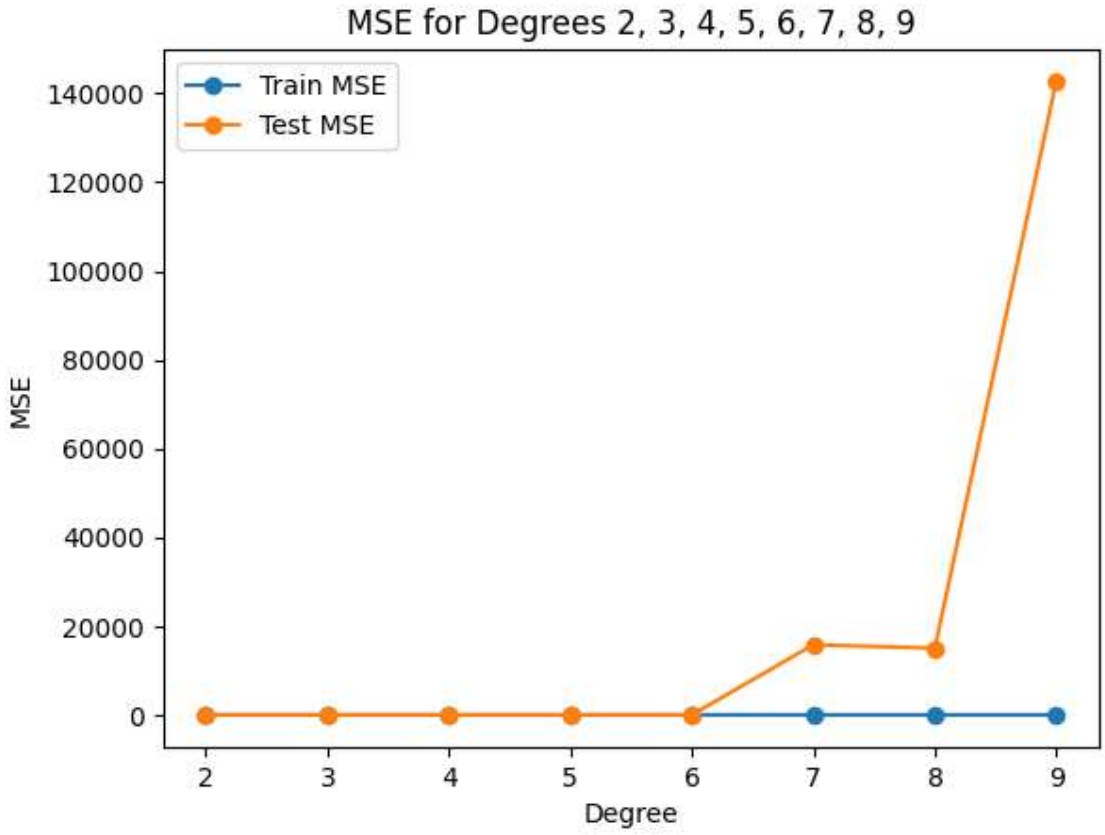
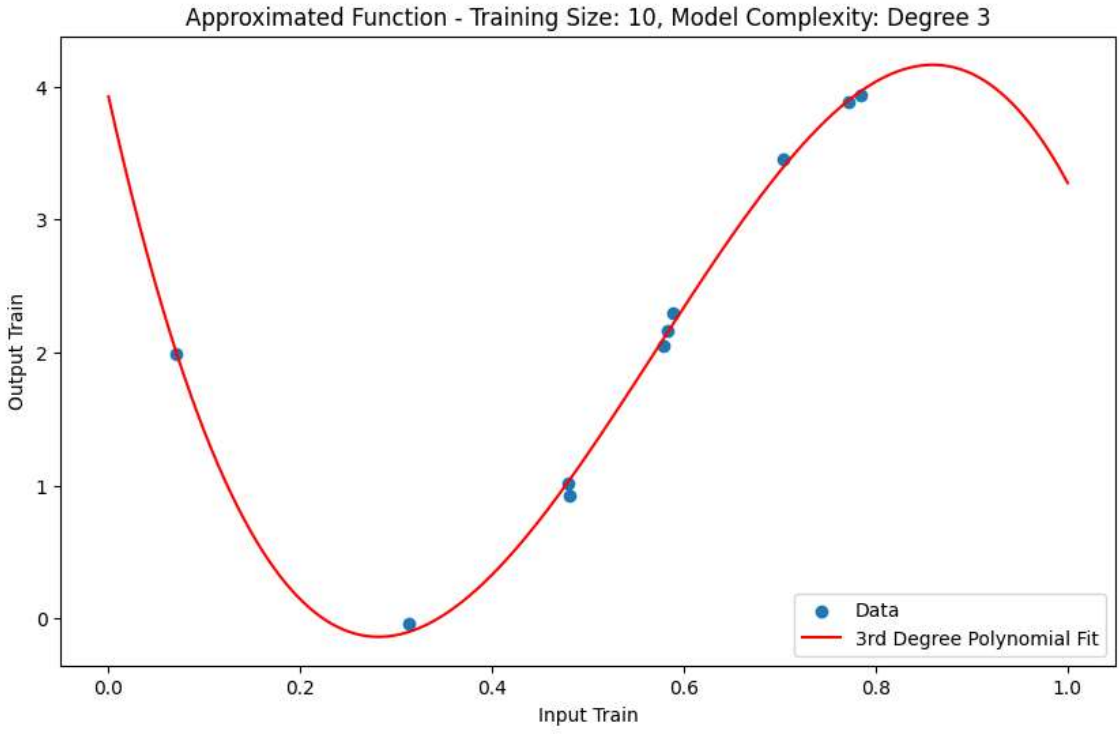


Fig88 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

2. **For degree/model complexity being 3**



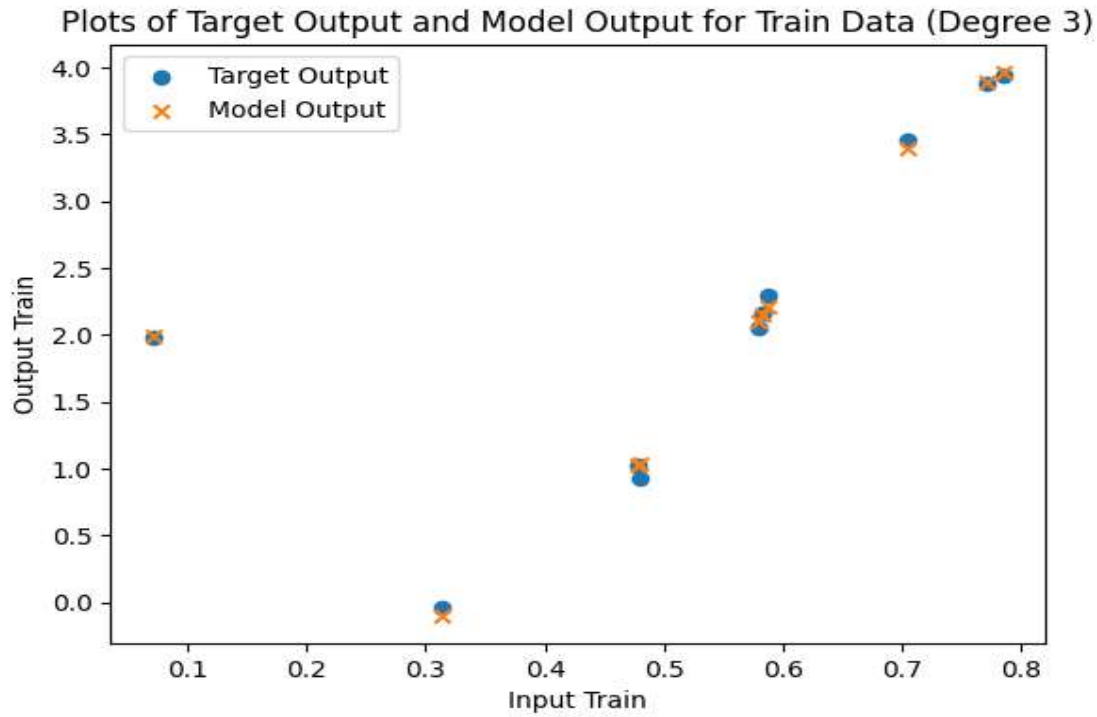


Fig89 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 3)

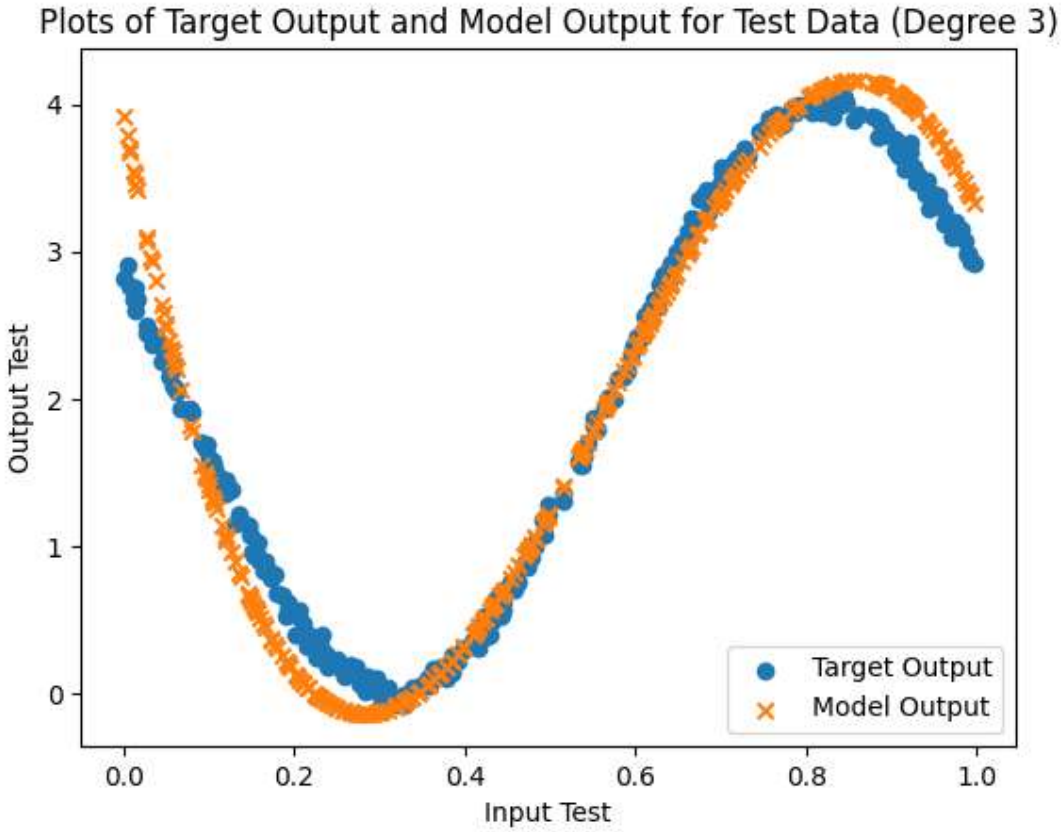


Fig90 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 3).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 3:	3.9235260496949182
w1 value of degree 3:	-32.38515321923802
w2 value of degree 3:	76.33981477296648
w3 value of degree 3:	-44.60477049000846
MSE for degree 3 training data:	0.0029394772850237292
MSE Percent for degree 3 training data:	0.1356512948429741
MSE for degree 3 test data:	0.08229764960415377
MSE Percent for degree 3 test data:	4.114626238099496

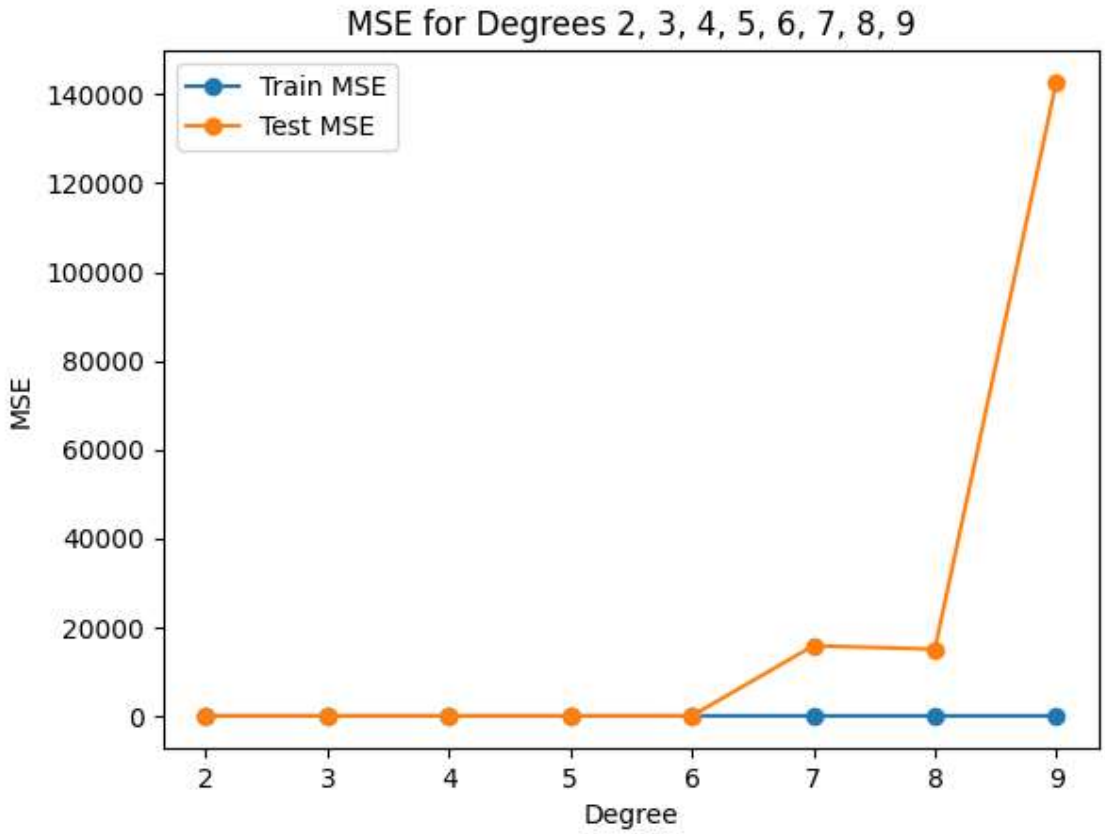


Fig91 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

3. **For degree/model complexity being 4**

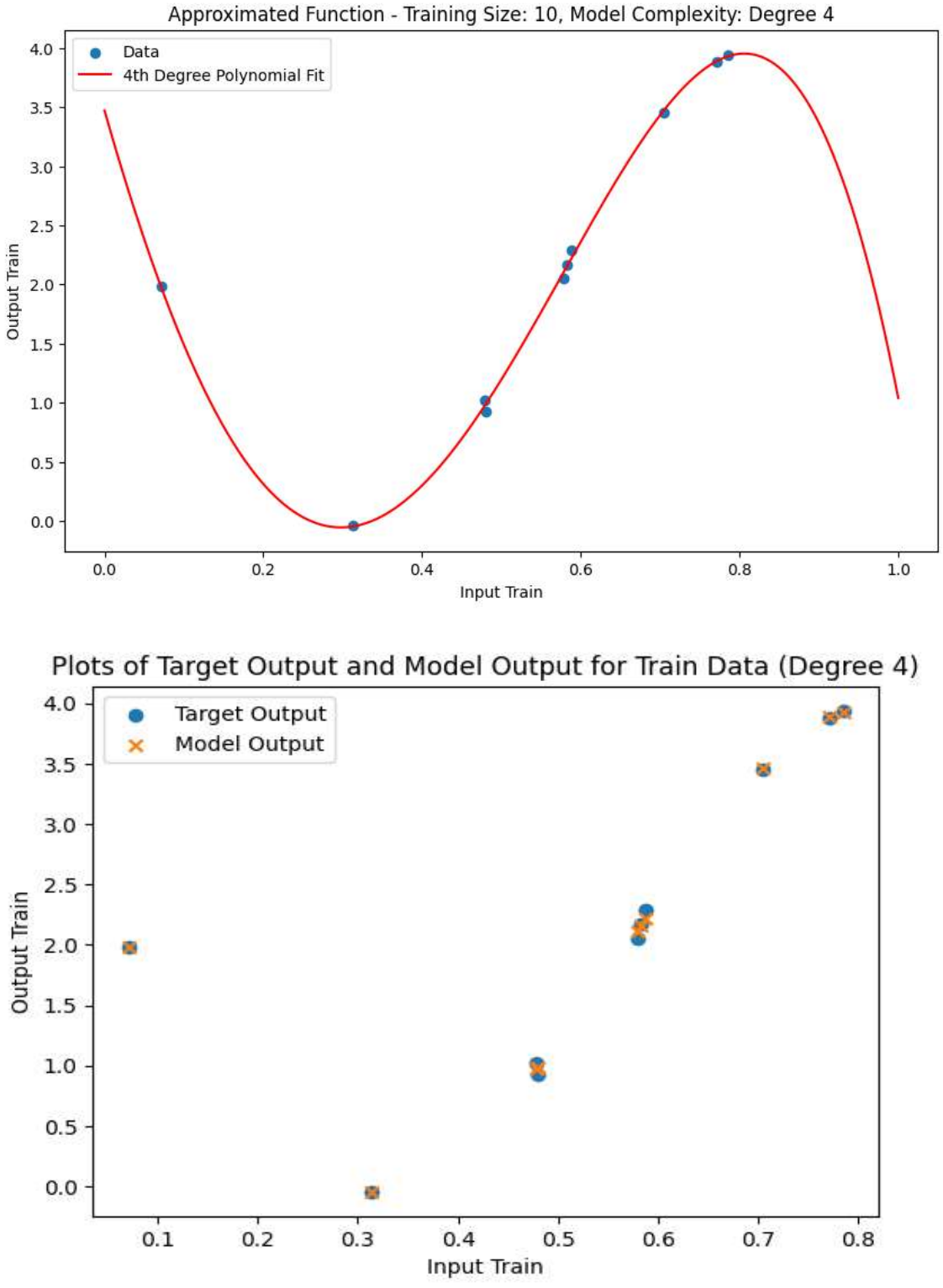


Fig92 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 4).

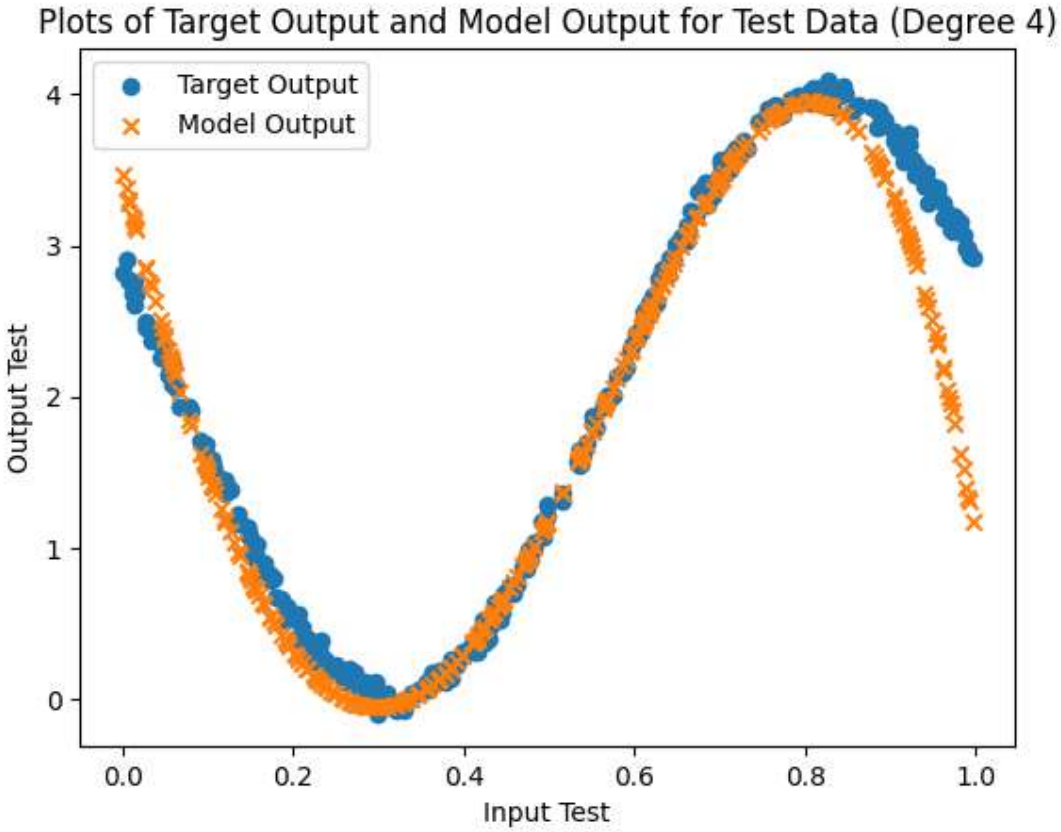


Fig93 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 4).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 4:	3.470994062000291
w1 value of degree 4:	-23.585800519693443
w2 value of degree 4:	35.735831332884516
w3 value of degree 4:	23.922356767162142
w4 value of degree 4:	-38.50343785350566
MSE for degree 4 training data:	0.0014694298595570516
MSE Percent for degree 4 train data:	0.06781139767447959
MSE for degree 4 test data:	0.14007290540685502
MSE Percent for degree 4 test data:	7.003209139095325

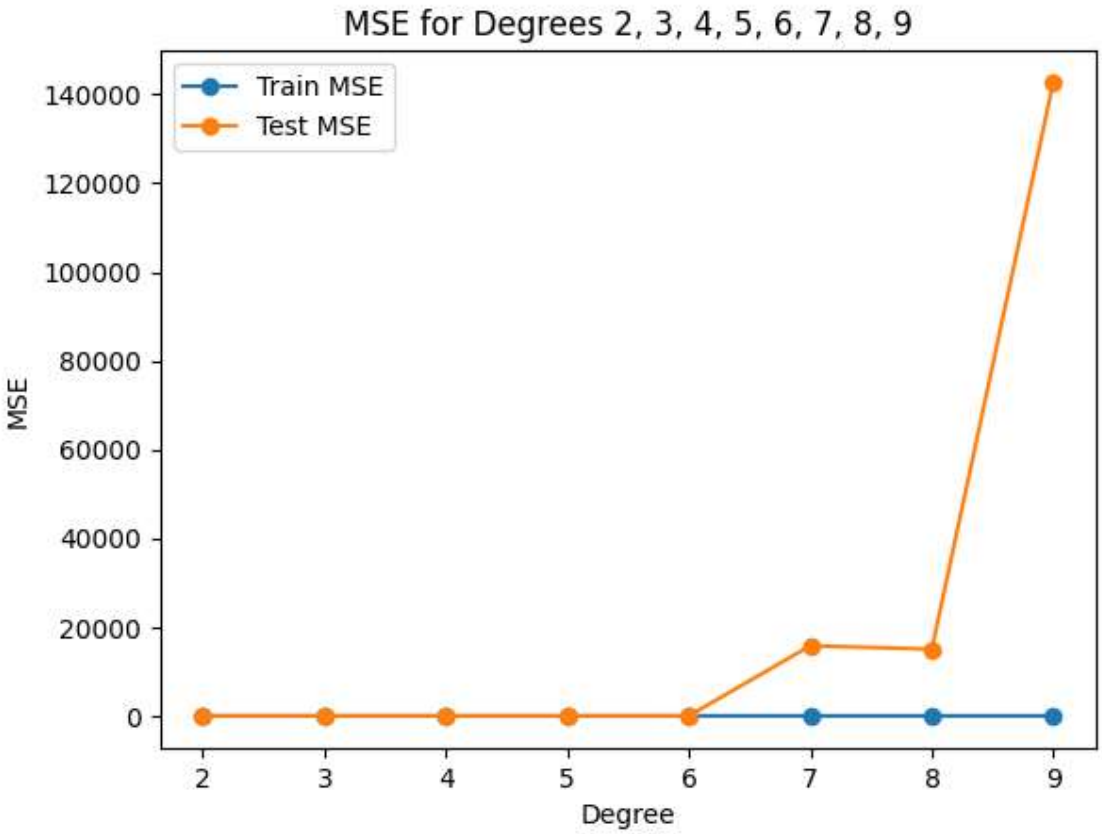


Fig94 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

4. **For degree/model complexity being 5**

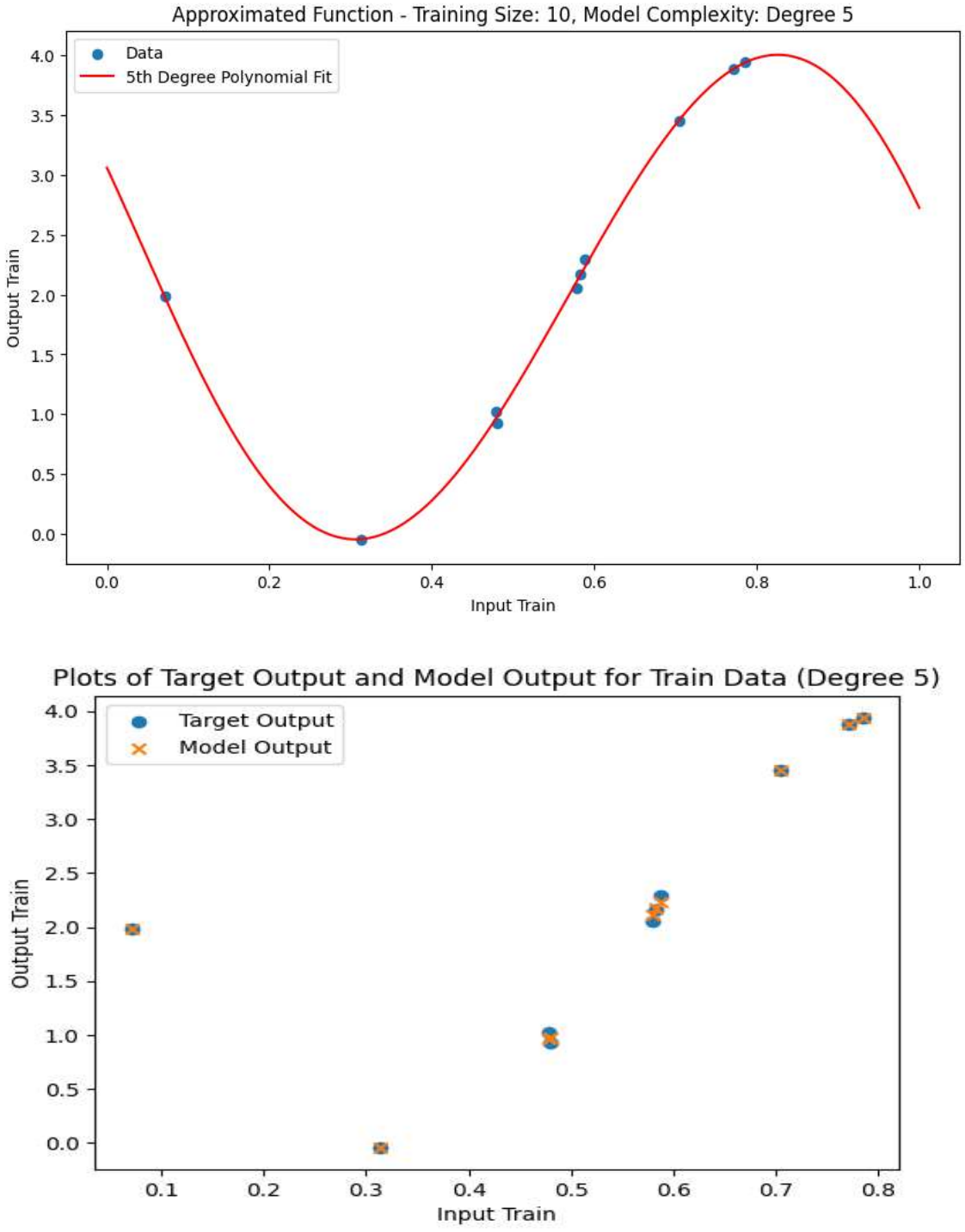


Fig95 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 5)

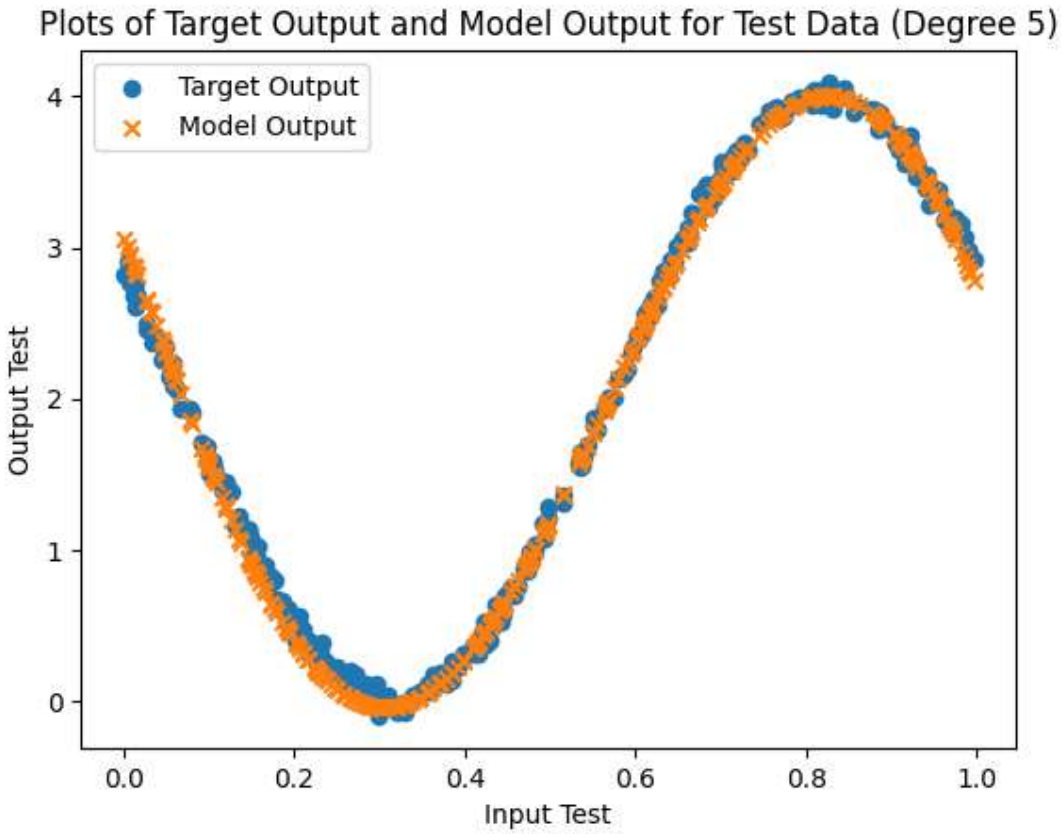


Fig96 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 5).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 5:	3.05994345988822
w1 value of degree 5:	-14.561287539677984
w2 value of degree 5:	-19.08370890149376
w3 value of degree 5:	164.46091342562275
w4 value of degree 5:	-200.80622735631556
w5 value of degree 5:	69.65528266336432
MSE for degree 5 training data:	0.0014123897680916743
MSE Percent for degree 5 train data:	0.06517910576847914
MSE for degree 5 test data:	0.007516066611749577
MSE Percent for degree 5 test data:	0.3757799285491075

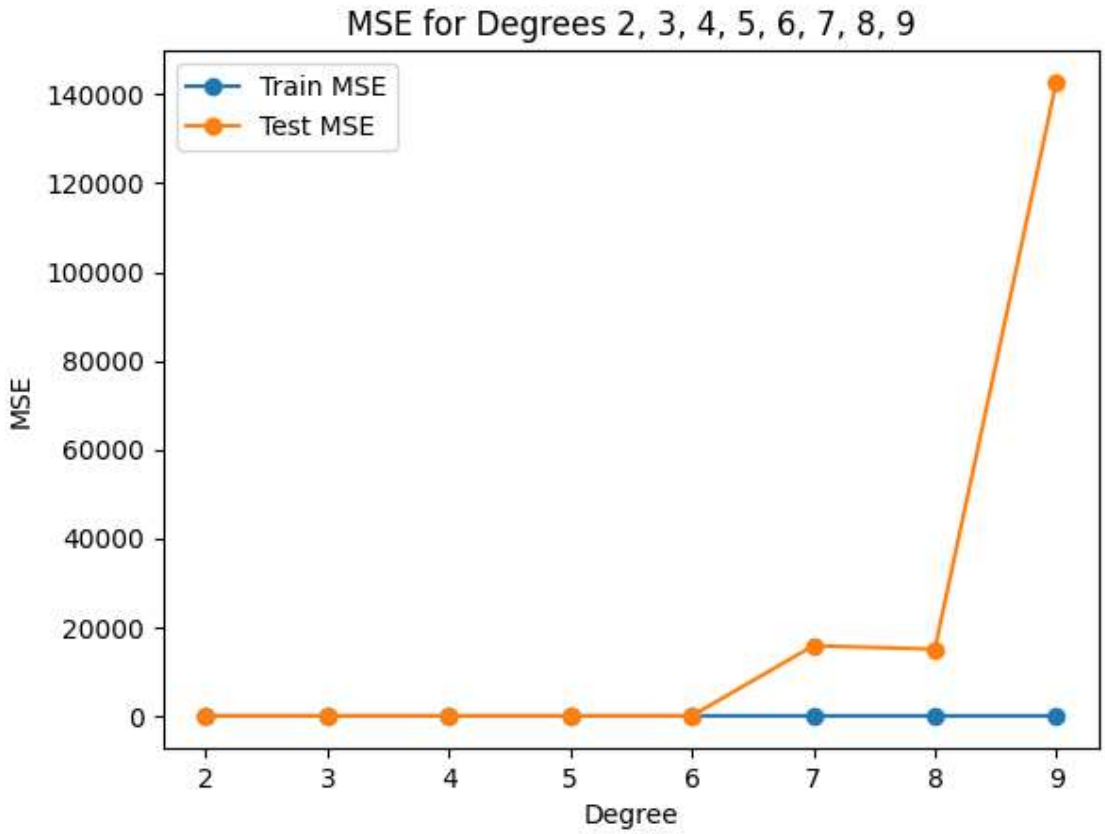


Fig97 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

5. **For degree/model complexity being 6**

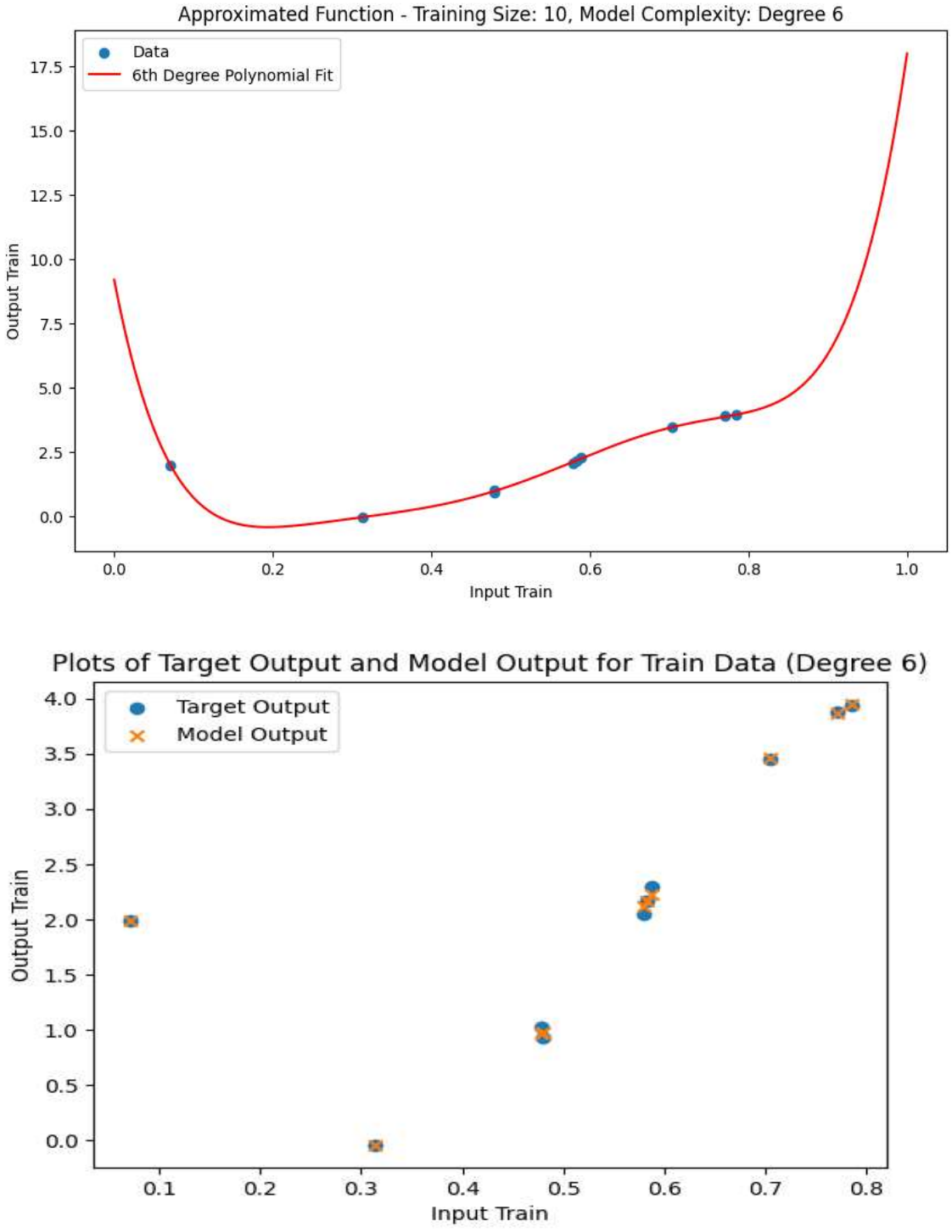


Fig98 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 6)

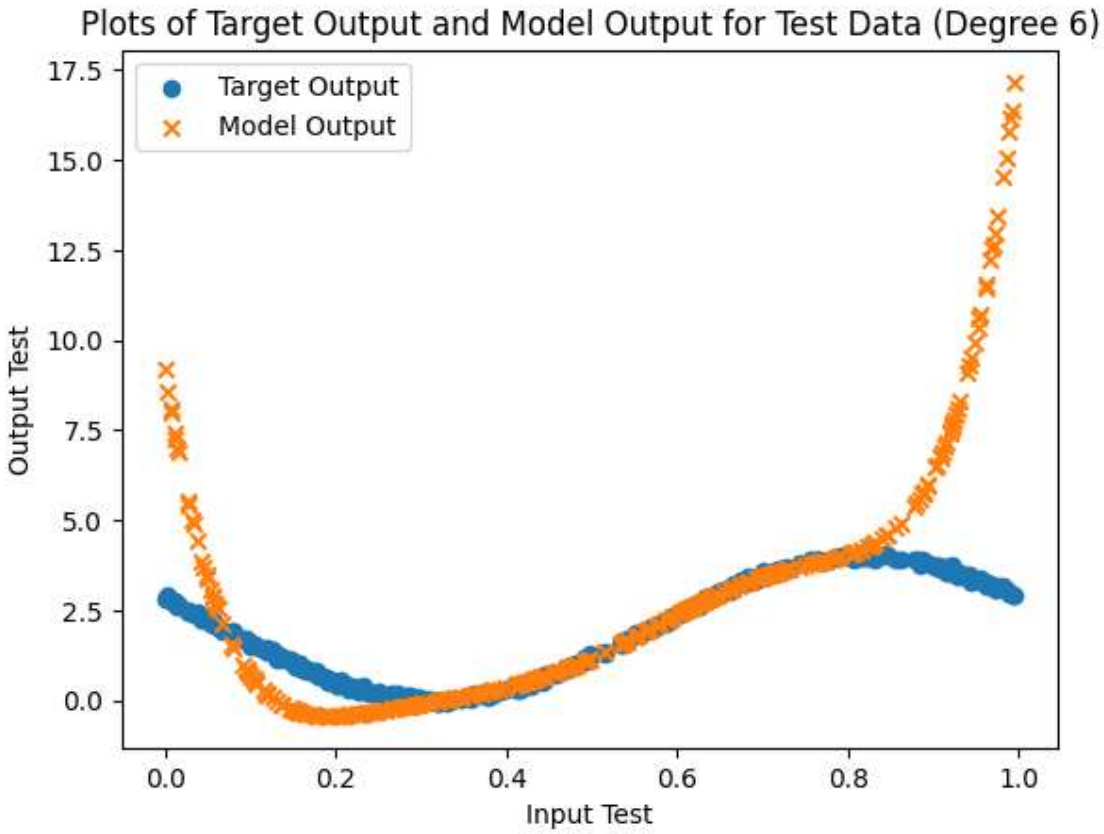


Fig99 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 6).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 6:	9.189597280798653
w1 value of degree 6:	-160.14789559445998
w2 value of degree 6:	1037.330959190389
w3 value of degree 6:	-3364.376800515255
w4 value of degree 6:	5849.63694486737
w5 value of degree 6:	-5115.171541579752
w6 value of degree 6:	1761.5310848587708
MSE for degree 6 training data:	0.0013619274300224443
MSE Percent for degree 6 training data:	0.06285036469101929
MSE for degree 6 test data:	8.217195951957521
MSE Percent for degree 6 test data:	410.83421252194387

We observe that:-

- Degree 6 is a better fit than degree 5.
- Both train and test MSE have reduced from degree 5 to 6 but not as drastically as they did from degree 4 to 5 and the reduction is extremely minute.

What we observe is that the values **of our parameters have increased significantly from degree 5 to 6**

That being said, our test MSE is high since we have very few-number of training samples but the model has lost its generalisation ability.

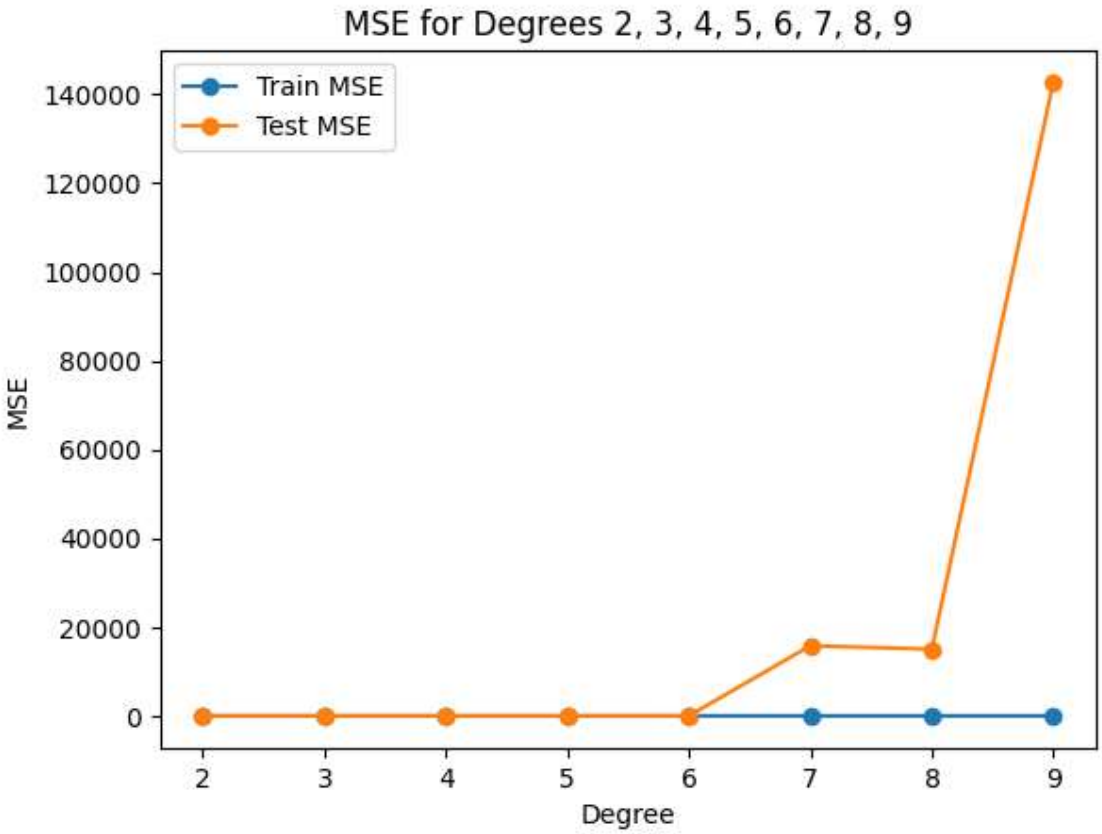


Fig100 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

6. **For degree/model complexity being 7**

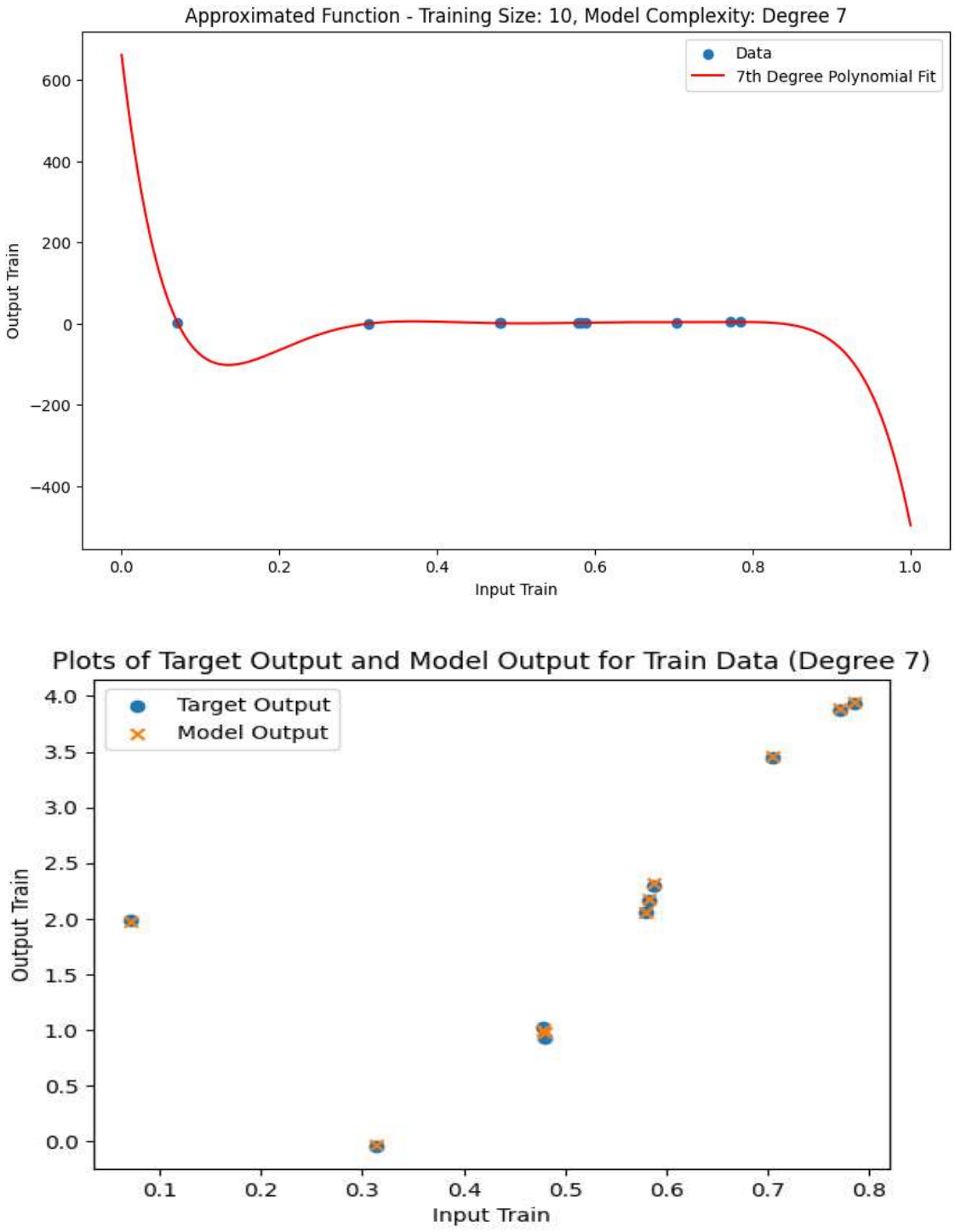


Fig101 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 7)

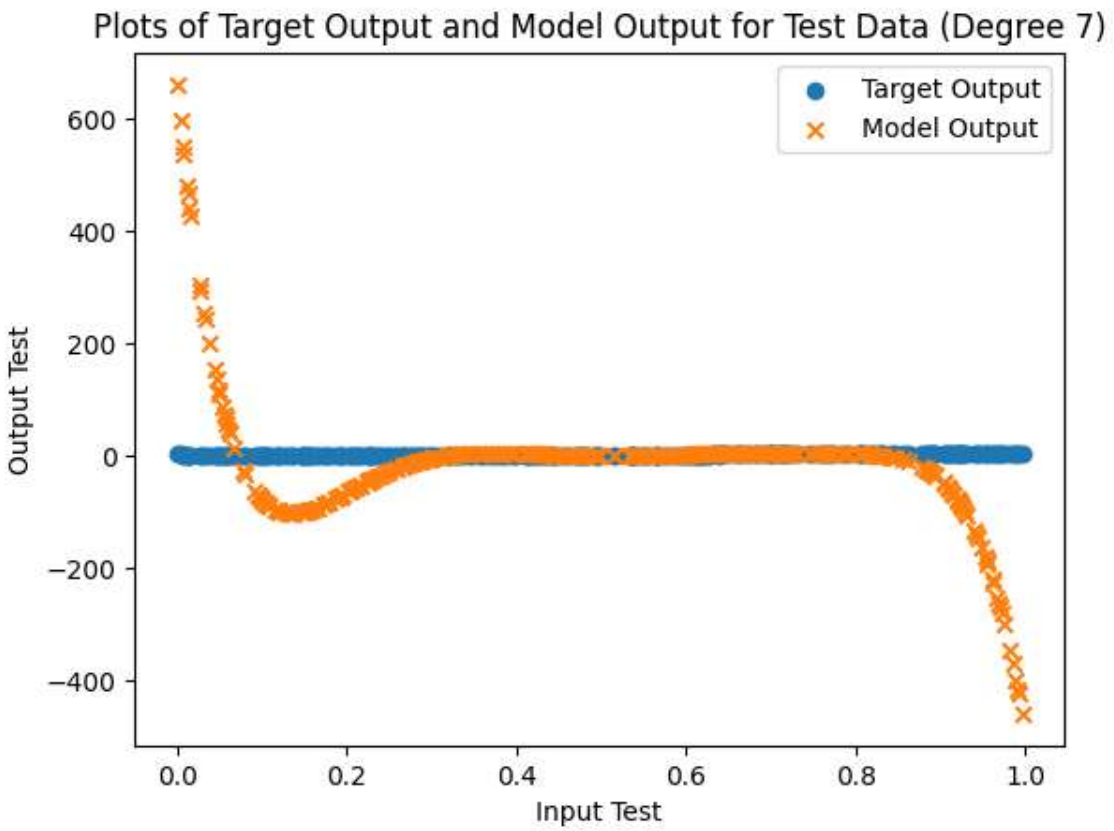


Fig102 **Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 7).**

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 7: 661.2194233105813

w1 value of degree 7: -16514.17933453462

w2 value of degree 7: 134006.03594925953

w3 value of degree 7: -528069.090907625

w4 value of degree 7: 1147882.1579622647

w5 value of degree 7: -1410531.8200313332

w6 value of degree 7: 920269.6028747887

w7 value of degree 7: -248199.66743487303

MSE for degree 7 training data: 0.00038551116517127477

MSE Percent for degree 7 train data: 0.017790608214033176

MSE for degree 7 test data: 15798.824106349884

MSE Percent for degree 7 test data: 789892.014070655

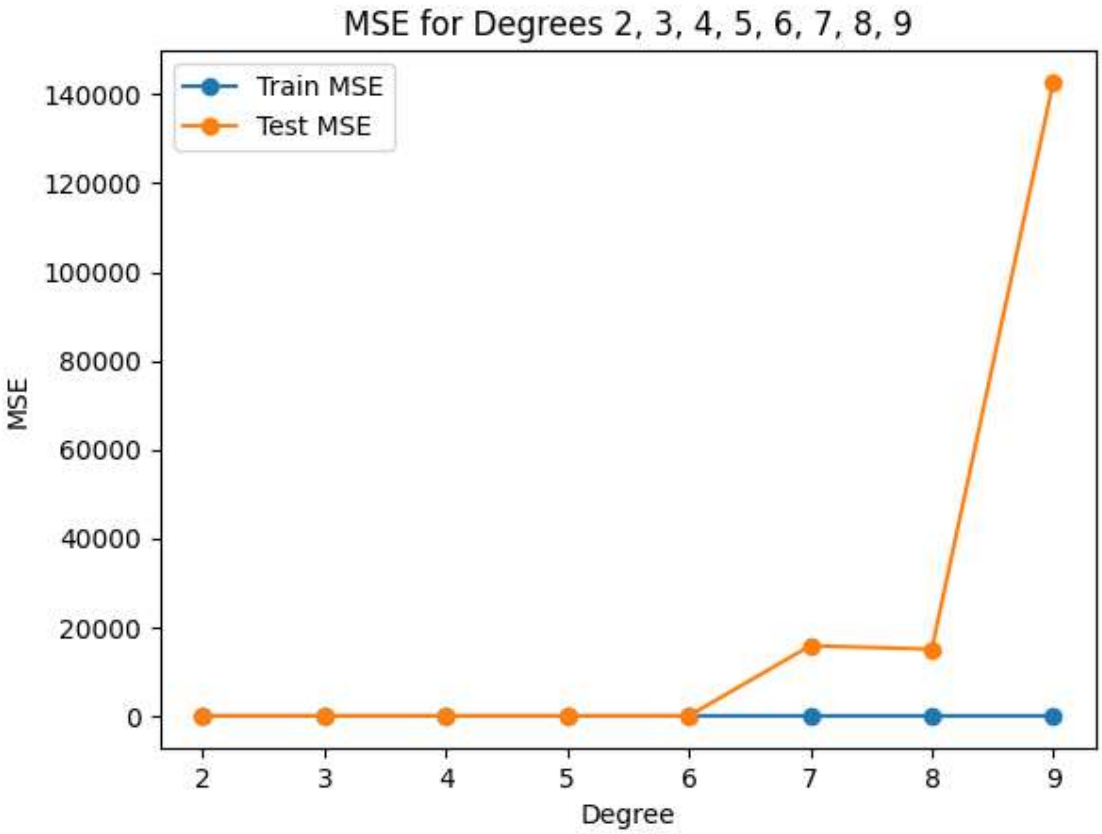


Fig103 Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.

7. **For degree/model complexity being 8**

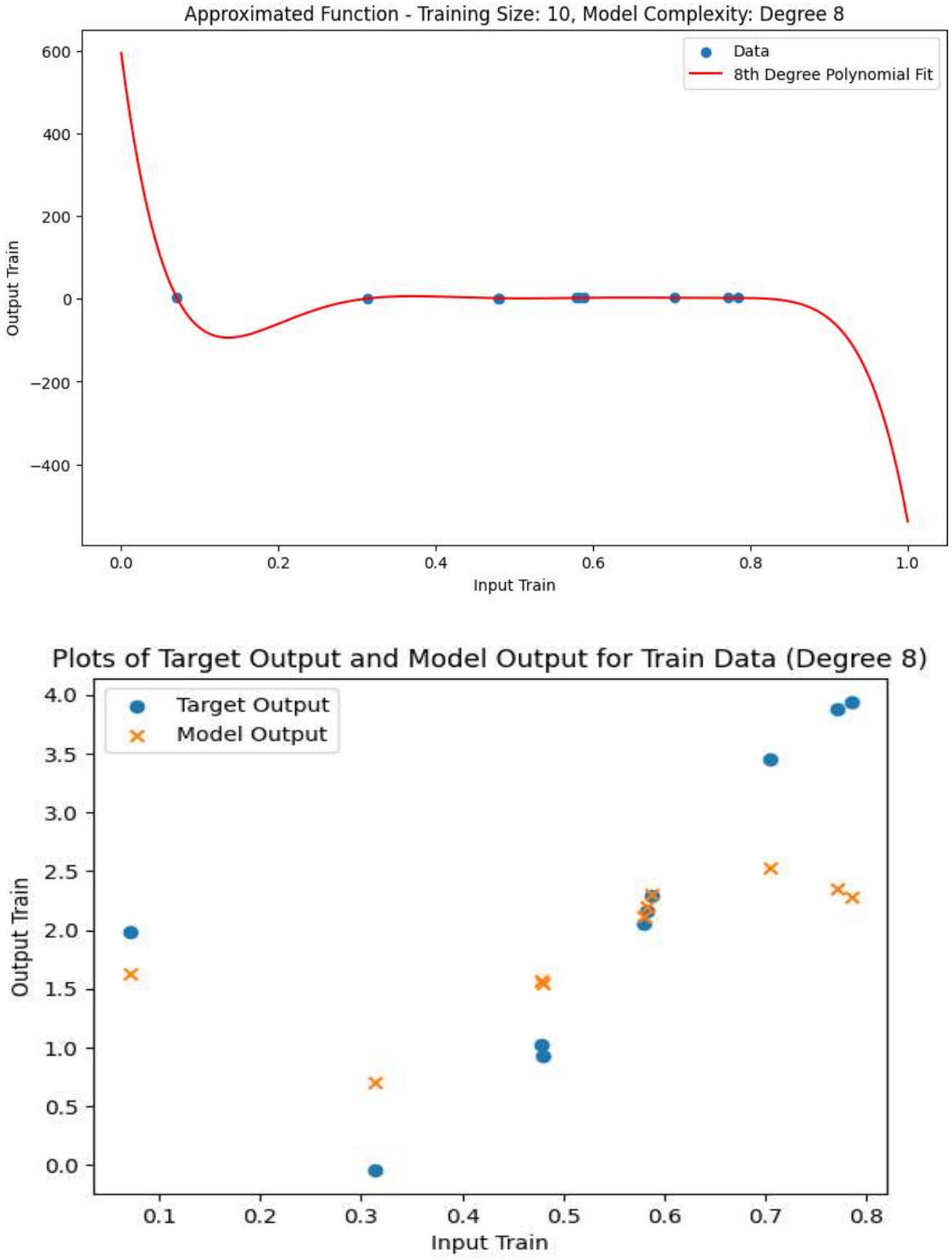


Fig104 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 8)

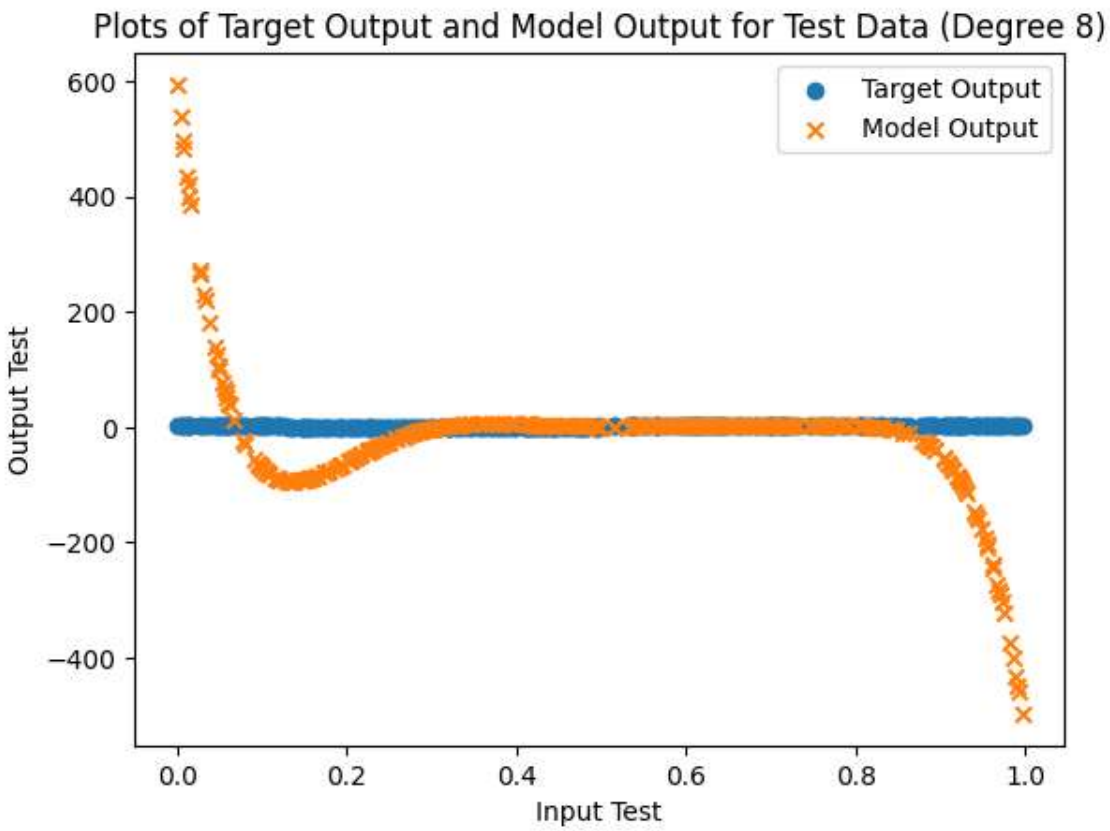


Fig105 **Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 8).**

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 8:	594.2141609971521
w1 value of degree 8:	-14731.443545146203
w2 value of degree 8:	117599.53323687114
w3 value of degree 8:	-451171.40324714105
w4 value of degree 8:	938622.3618639546
w5 value of degree 8:	-1064768.026673146
w6 value of degree 8:	577191.911177756
w7 value of degree 8:	-59788.76334621701
w8 value of degree 8:	-44085.941592659074
MSE for degree 8 training data:	0.7307280313531523
MSE Percent for degree 8 train data:	33.72171104574884
MSE for degree 8 test data:	15016.05242810468
MSE Percent for degree 8 test data:	750755.8673976844

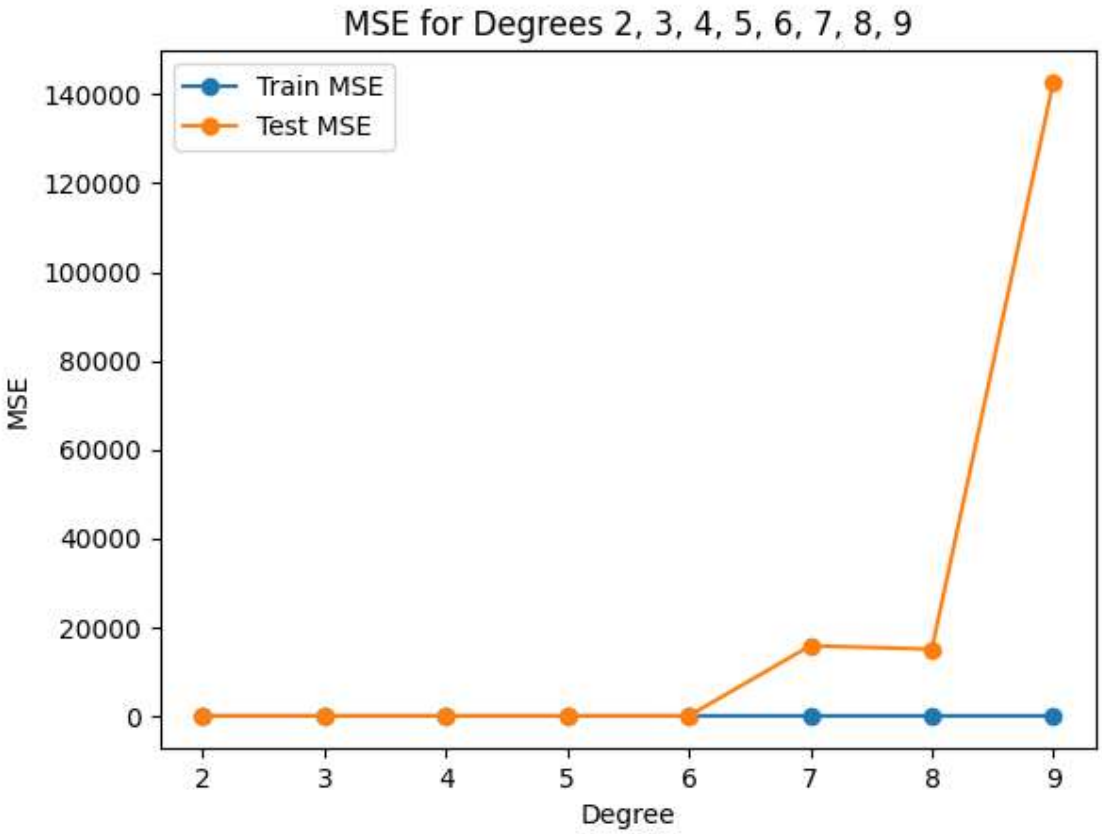
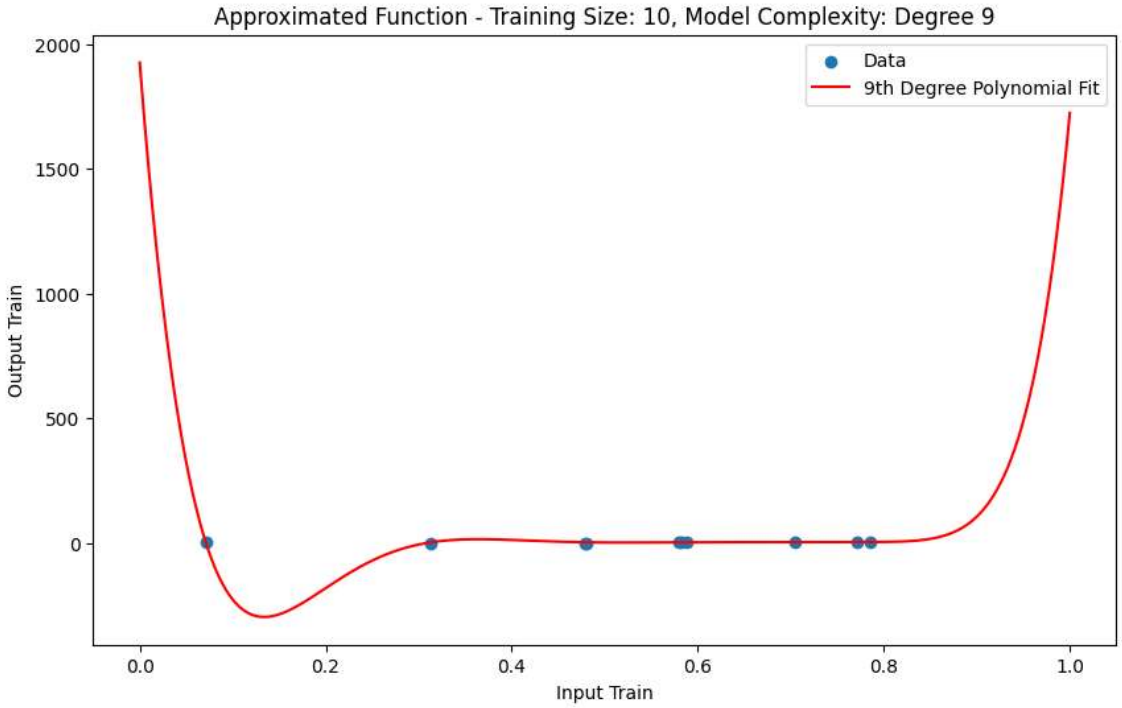


Fig106 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

8. **For degree/model complexity being 9**



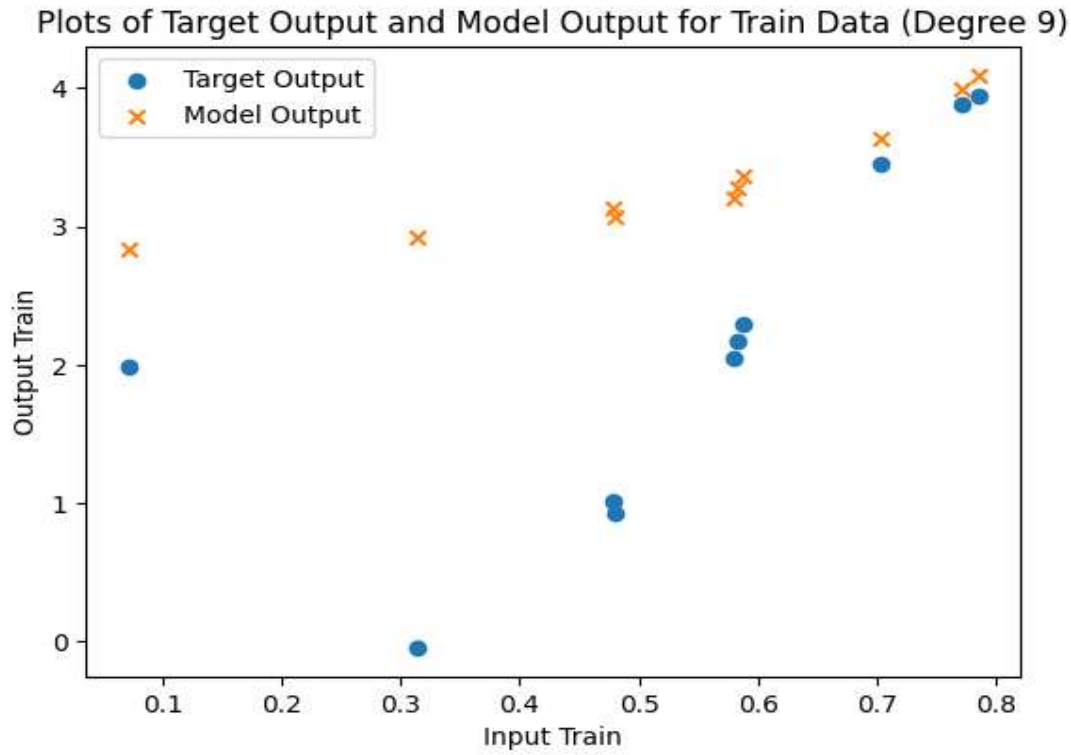


Fig107 Plots of Target Output/Actual output and Model Output/Predicted output for Train Data (Degree 9)

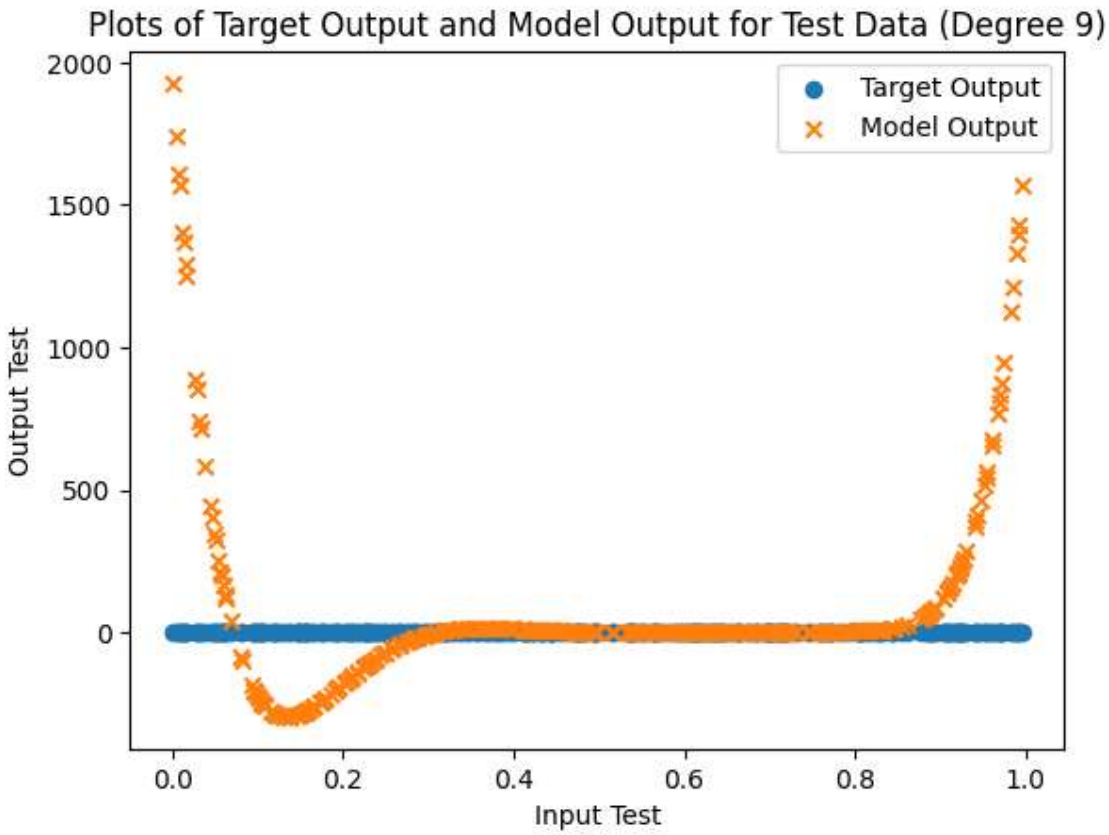


Fig108 Plots of Target Output/Actual output and Model Output/Predicted output for Test Data (Degree 9).

Values of the coefficients and train and test MSE and MSE percent values.

w0 value of degree 9: 1926.5237413047523

w1 value of degree9 : -47658.99585474458

w2 value of degree 9: 372791.2001995645

w3 value of degree 9: -1309652.1653685276

w4 value of degree 9: 1934633.8951865165

w5 value of degree 9: 674701.8625535371

w6 value of degree 9: -6805151.5890938165

w7 value of degree 9: 10242084.396416912

w8 value of degree 9: -6852725.395698675

w9 value of degree 9: 1790775.0638955447

MSE for degree 9 training data: 2.2338467838849048

MSE Percent for degree 9 train data: 103.0877871581697

MSE for degree 9 test data: 142760.72260319252

MSE Percent for degree 9 test data: 7137591.630119797

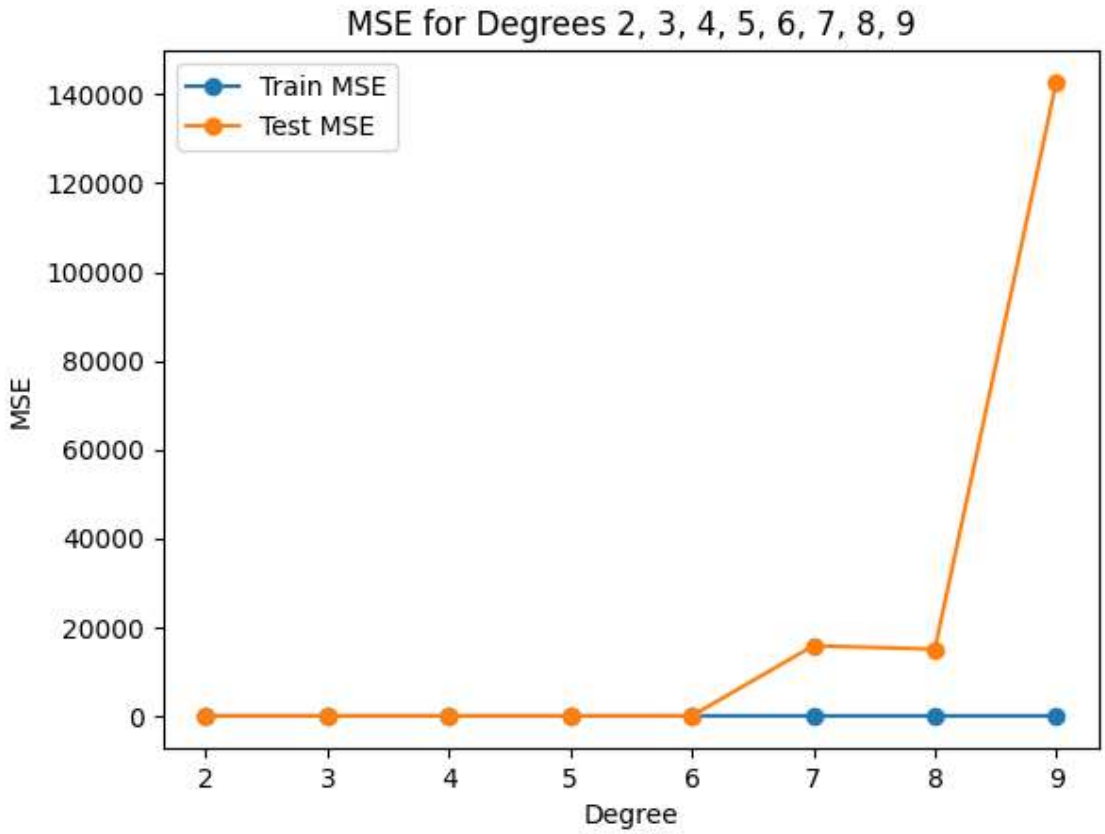


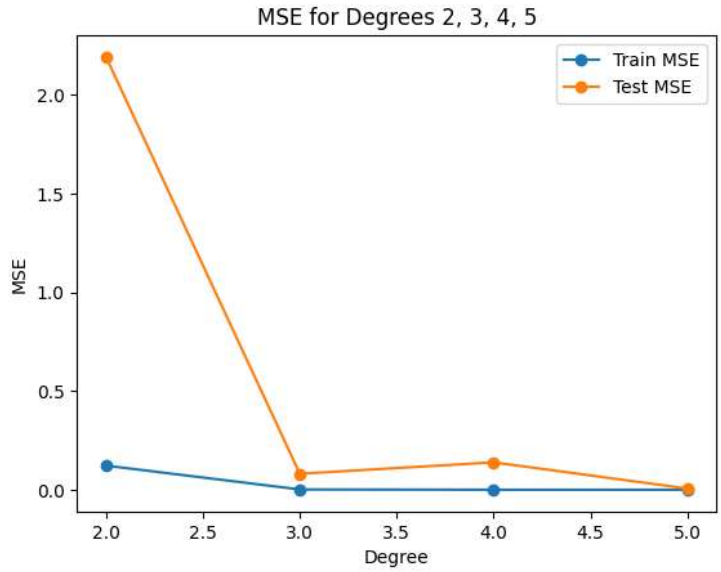
Fig109 **Plots of Train and Test MSE on y axis Vs Degrees on x axis for degrees 2, 3, 4, 5, 6, 7, 8 and 9.**

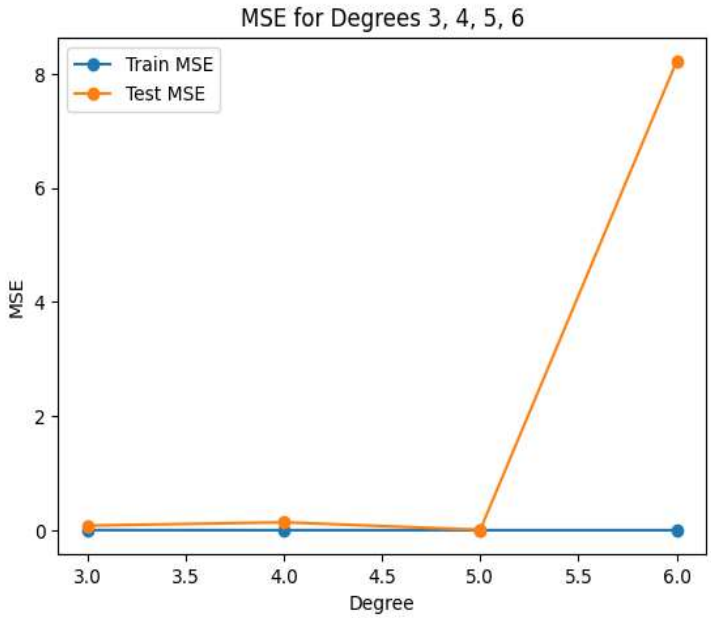
We observe that for degree 5 and above (including 6, 7, 8 and 9) the train and test MSE values are almost the same and we are just increasing the complexity of the model which is resulting in huge values of parameters.

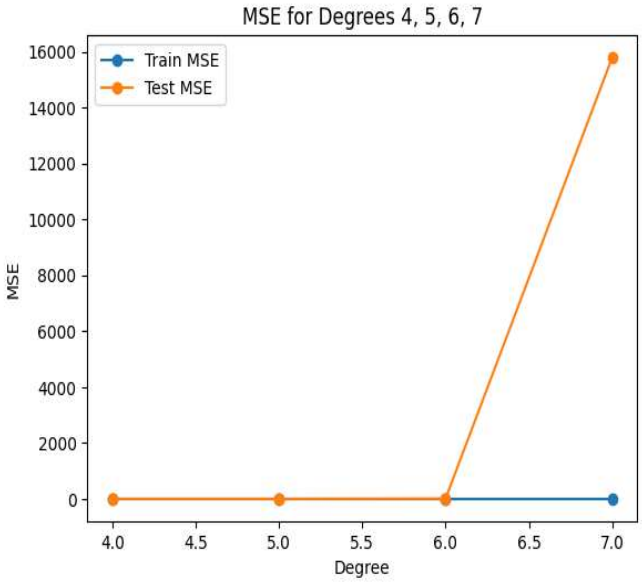
Regularization:

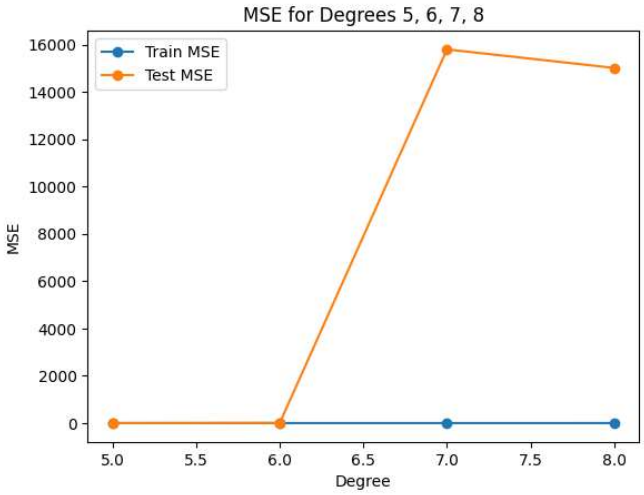
For regularization we take a close look at our MSE plots but we do it piecewise.

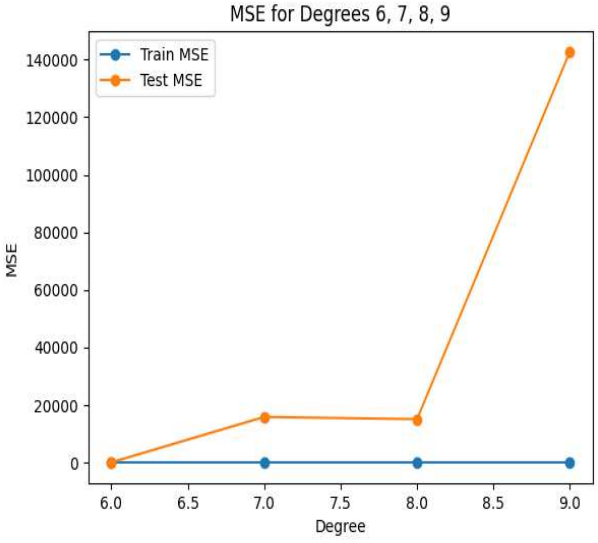
For 10 samples we have few piecewise MSE curves

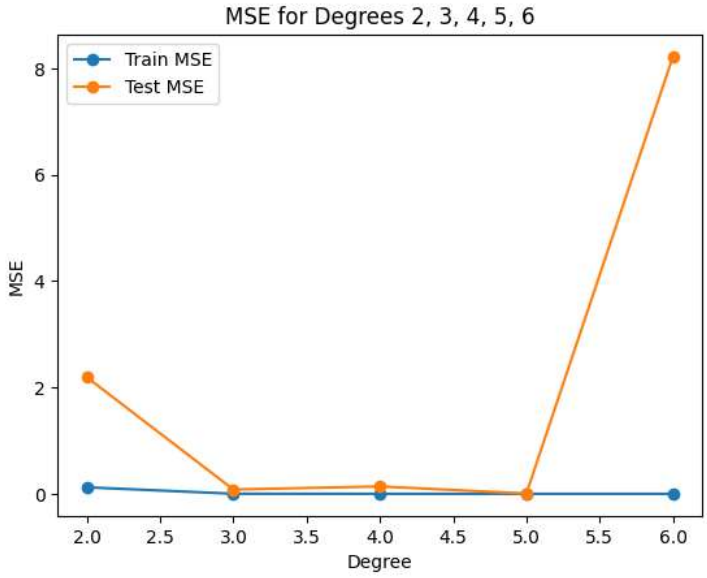


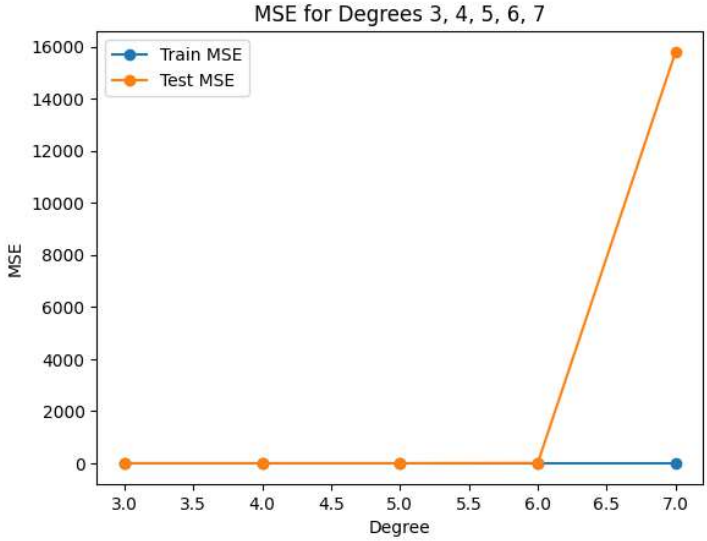


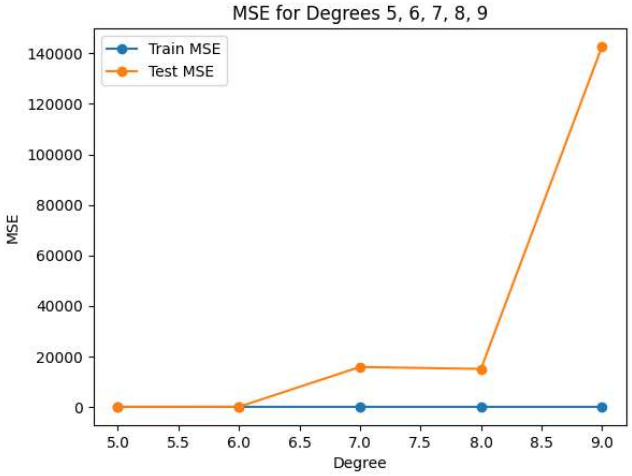

MSE for Degrees 4, 5, 6, 7
Train MSE
Test MSE
MSE
Degree
16000
14000
12000
10000
8000
6000
4000
2000
0
4.0
4.5
5.0
5.5
6.0
6.5
7.0

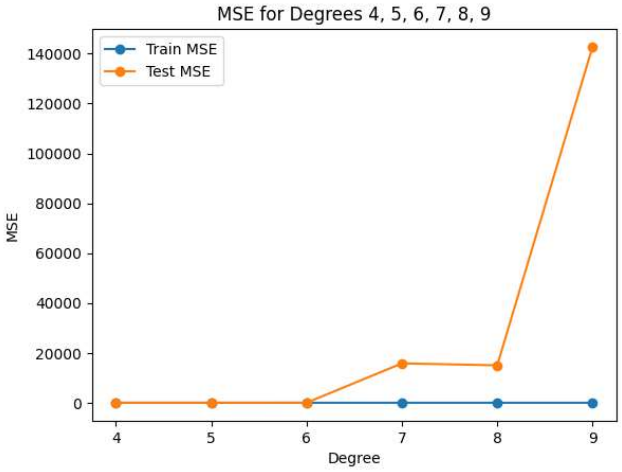

MSE for Degrees 5, 6, 7, 8
Train MSE
Test MSE
MSE
Degree
16000
14000
12000
10000
8000
6000
4000
2000
0
5.0
5.5
6.0
6.5
7.0
7.5
8.0

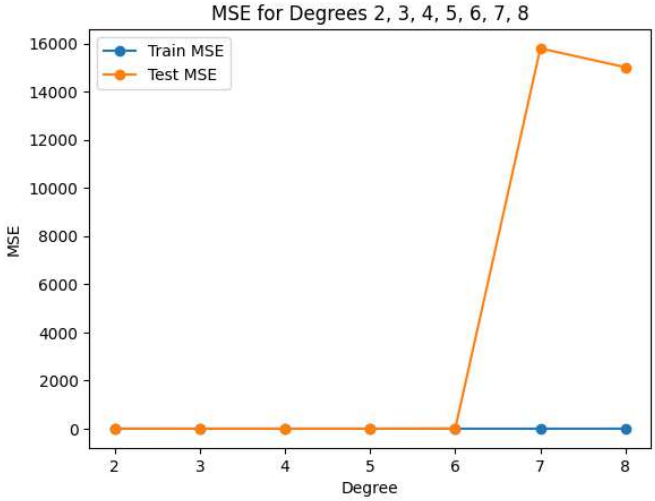

MSE for Degrees 6, 7, 8, 9
Train MSE
Test MSE
MSE
Degree
140000
120000
100000
80000
60000
40000
20000
0
6.0
6.5
7.0
7.5
8.0
8.5
9.0

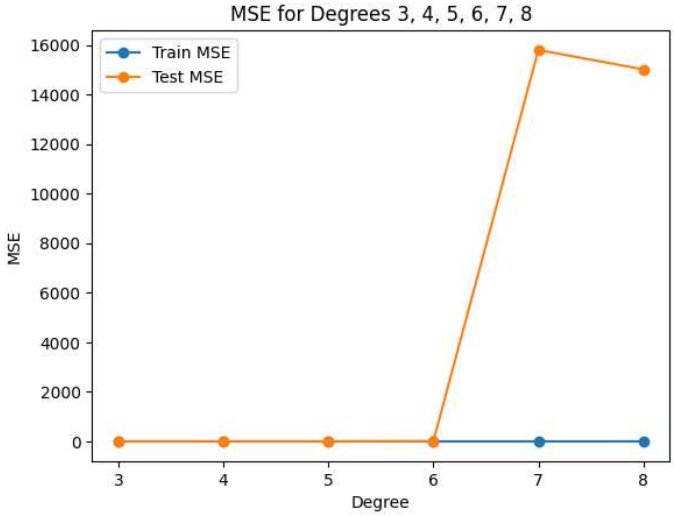

MSE for Degrees 2, 3, 4, 5, 6
Train MSE
Test MSE
MSE
Degree
0
2
4
6
8
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0

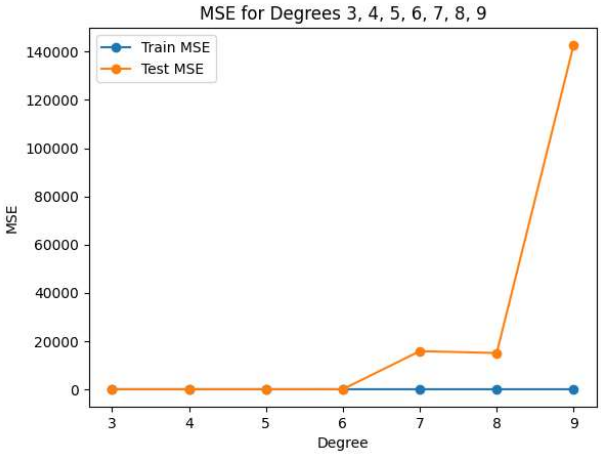

MSE for Degrees 3, 4, 5, 6, 7
Train MSE
Test MSE
MSE
Degree
0
2000
4000
6000
8000
10000
12000
14000
16000
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0

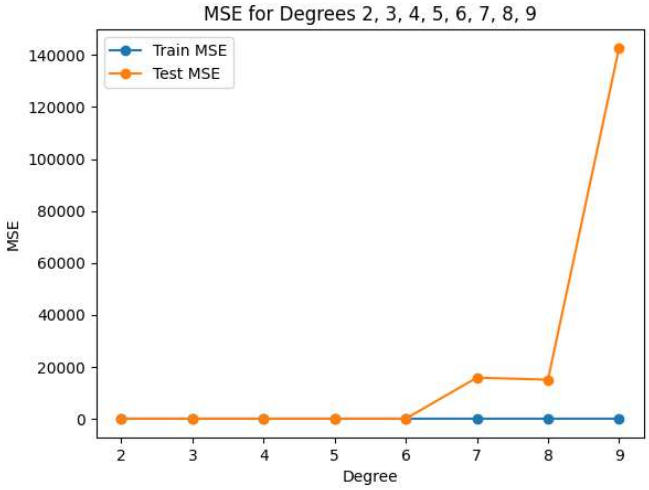

MSE for Degrees 5, 6, 7, 8, 9
Train MSE
Test MSE
MSE
Degree
0
20000
40000
60000
80000
100000
120000
140000
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


MSE for Degrees 4, 5, 6, 7, 8, 9
Train MSE
Test MSE
MSE
Degree
0
20000
40000
60000
80000
100000
120000
140000
4
5
6
7
8
9

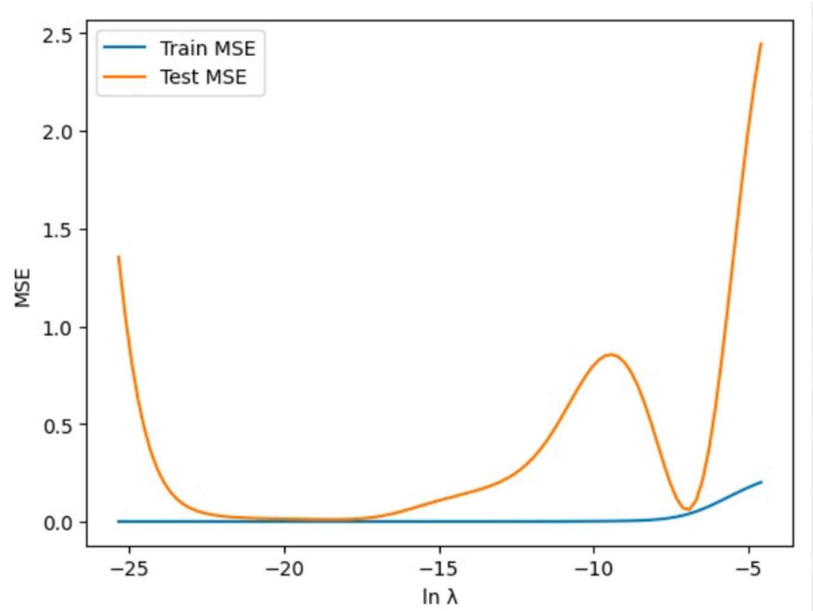

MSE for Degrees 2, 3, 4, 5, 6, 7, 8
Train MSE
Test MSE
MSE
Degree
0
2000
4000
6000
8000
10000
12000
14000
16000
2
3
4
5
6
7
8


MSE for Degrees 3, 4, 5, 6, 7, 8
Train MSE
Test MSE
MSE
Degree
0
2000
4000
6000
8000
10000
12000
14000
16000
3
4
5
6
7
8

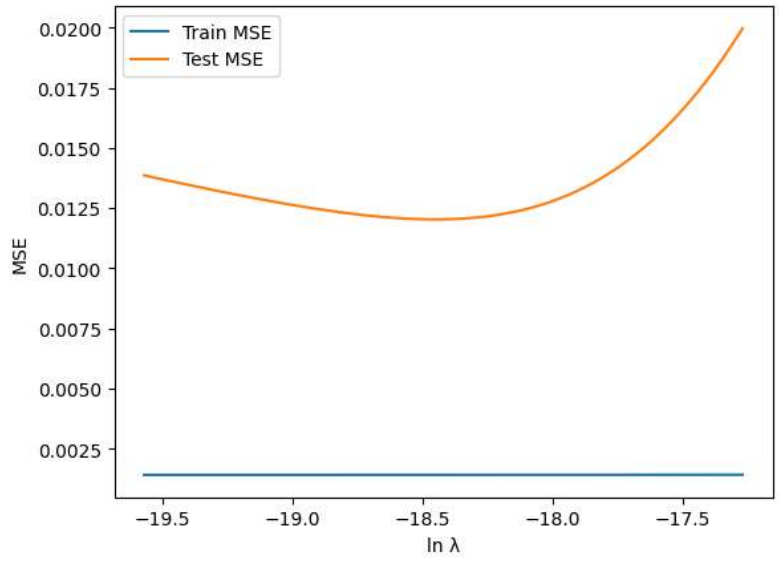

MSE for Degrees 3, 4, 5, 6, 7, 8, 9
Train MSE
Test MSE
MSE
Degree
0
20000
40000
60000
80000
100000
120000
140000
3
4
5
6
7
8
9

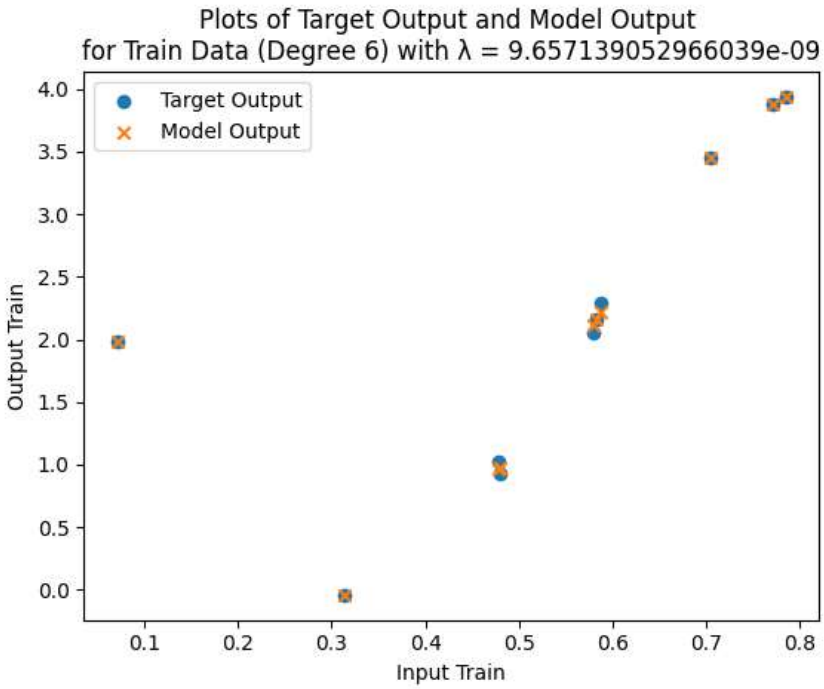

MSE for Degrees 2, 3, 4, 5, 6, 7, 8, 9
Train MSE
Test MSE
MSE
Degree
0
20000
40000
60000
80000
100000
120000
140000
2
3
4
5
6
7
8
9

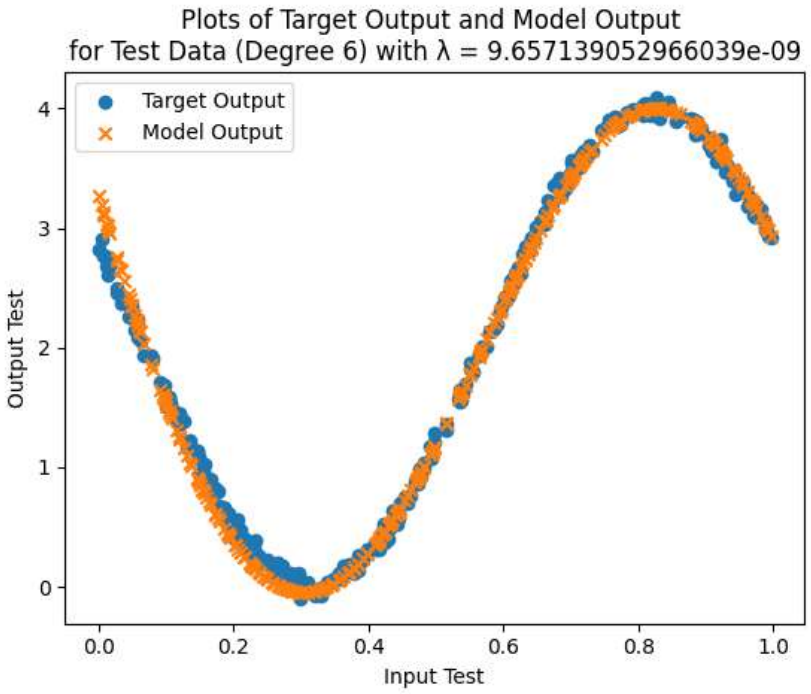
While regularization



degree 6 for 10 data 01:47




Plots of Target Output and Model Output
for Train Data (Degree 6) with λ = 9.657139052966039e-09
Target Output
Model Output
Output Train
Input Train
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Plots of Target Output and Model Output
for Test Data (Degree 6) with λ = 9.657139052966039e-09
Target Output
Model Output
Output Test
Input Test
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0

Before Regularisation:

w0 value of degree 6: 9.189597280798653

w1 value of degree 6: -160.14789559445998

w2 value of degree 6: 1037.330959190389

w3 value of degree 6: -3364.376800515255

w4 value of degree 6: 5849.63694486737

w5 value of degree 6: -5115.171541579752

w6 value of degree 6: 1761.5310848587708

MSE for degree 6 training data: 0.0013619274300224443

MSE Percent for degree 6 training data: 0.06285036469101929

MSE for degree 6 test data: 8.217195951957521

MSE Percent for degree 6 test data: 410.83421252194387

After Regularisation

w0 value of degree 6: 3.265498588558421

w1 value of degree 6: -19.342641310406293

w2 value of degree 6: 14.170204315636624

w3 value of degree 6: 59.054742365575315

w4 value of degree 6: -30.4290644648286

w5 value of degree 6: -67.32078540492883

w6 value of degree 6: 43.49372835632741

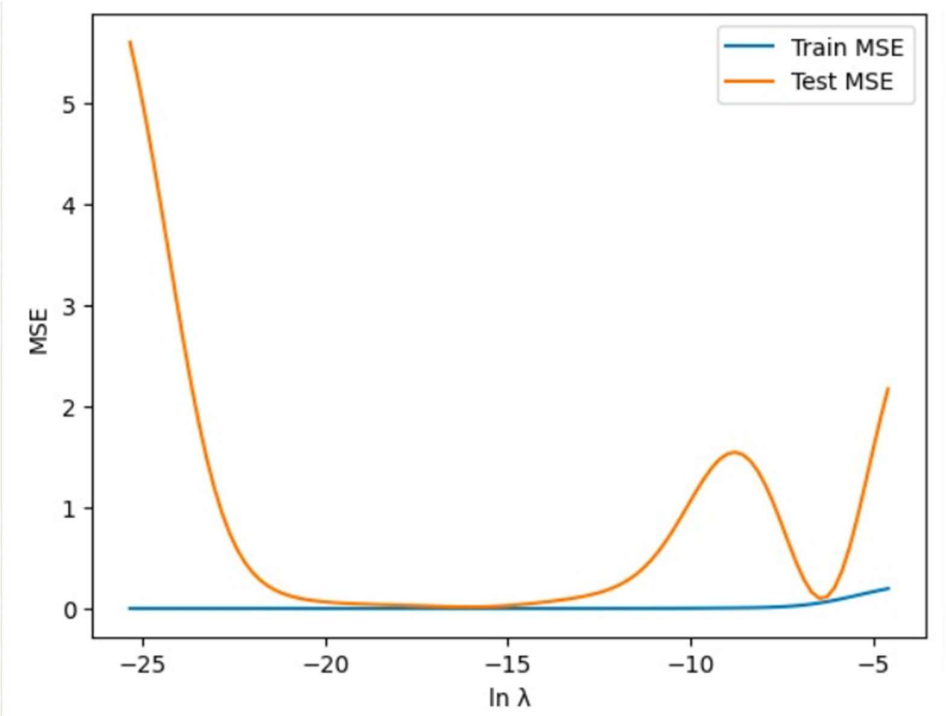
MSE for degree 6 training data: 0.0014108847430304582

MSE Percent for degree 6 training data: 0.06510965171984089

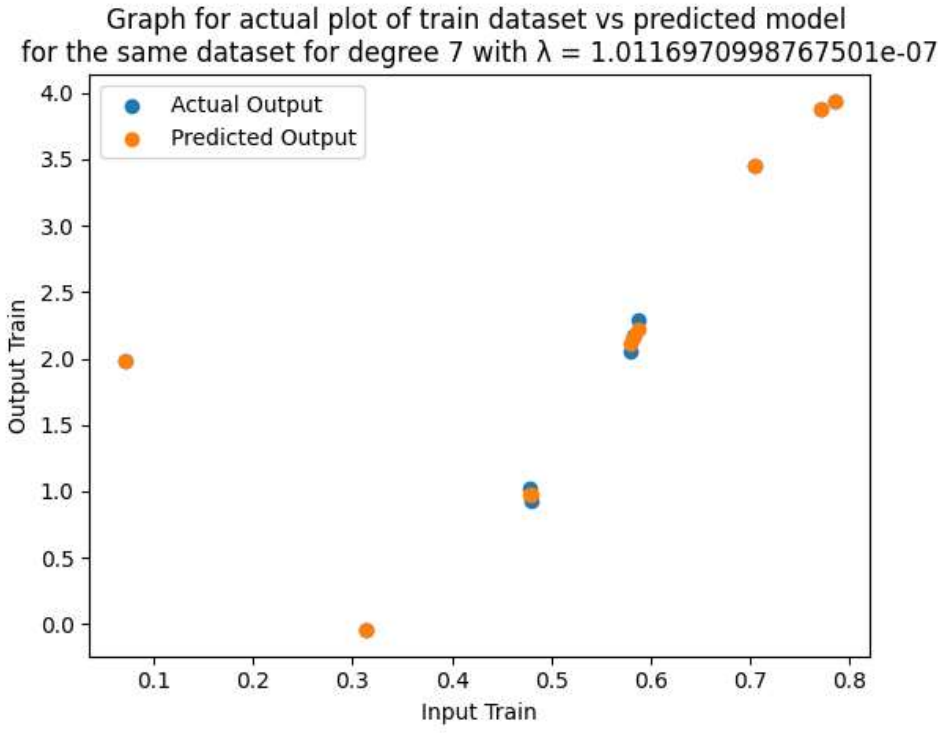
MSE for degree 6 test data: 0.012026675875635547

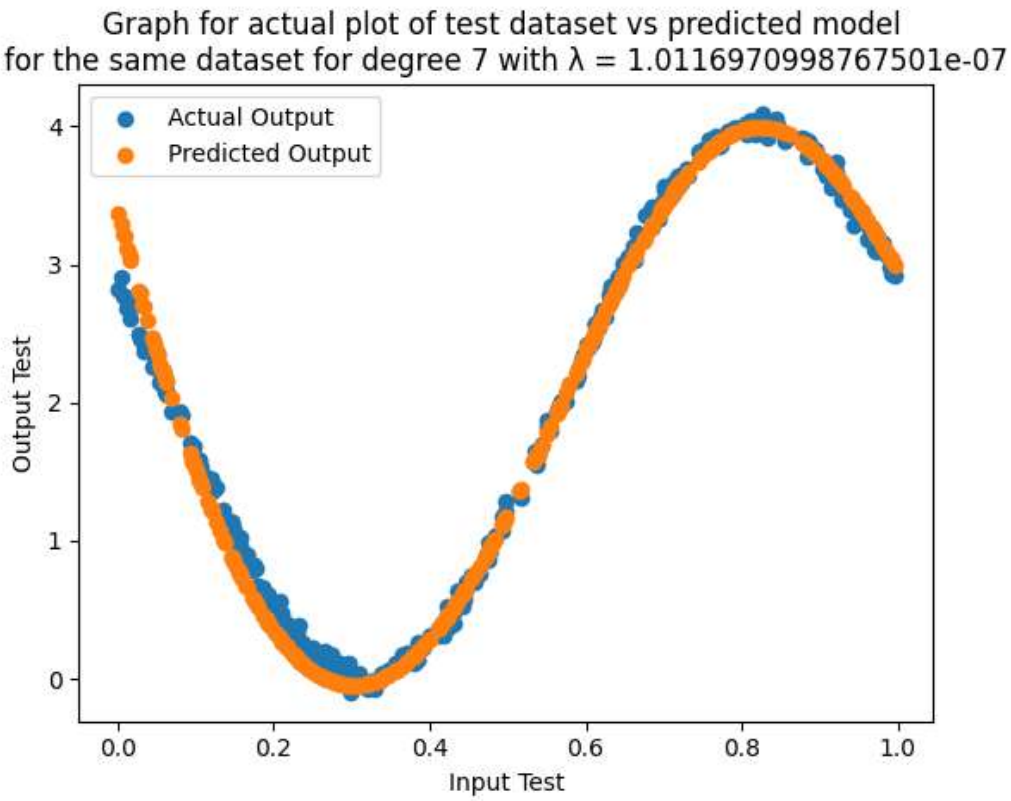
MSE Percent for degree 6 test data: 0.6012963475023788

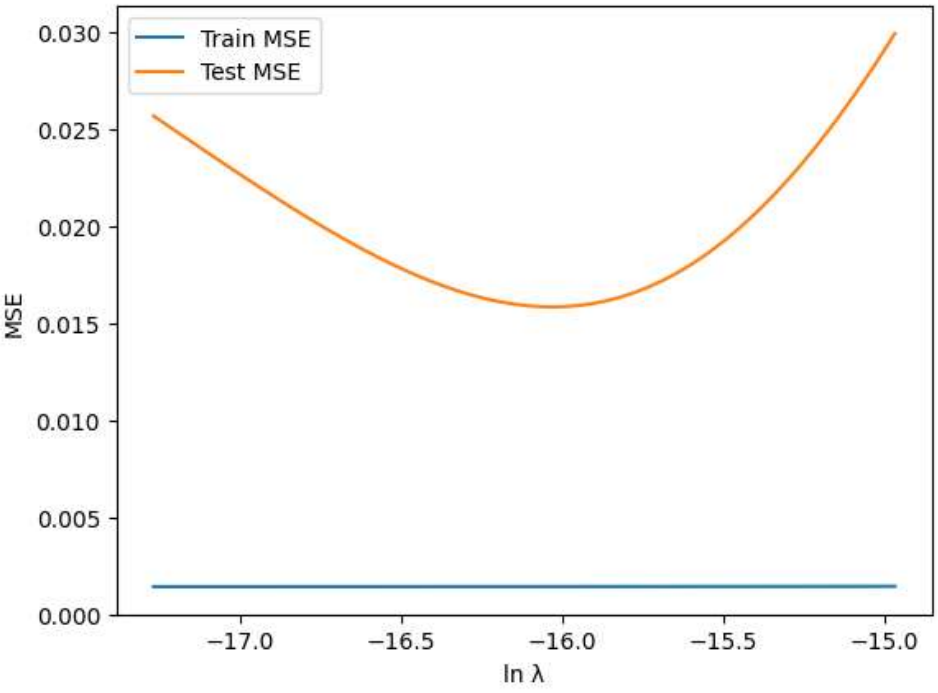
While regularization

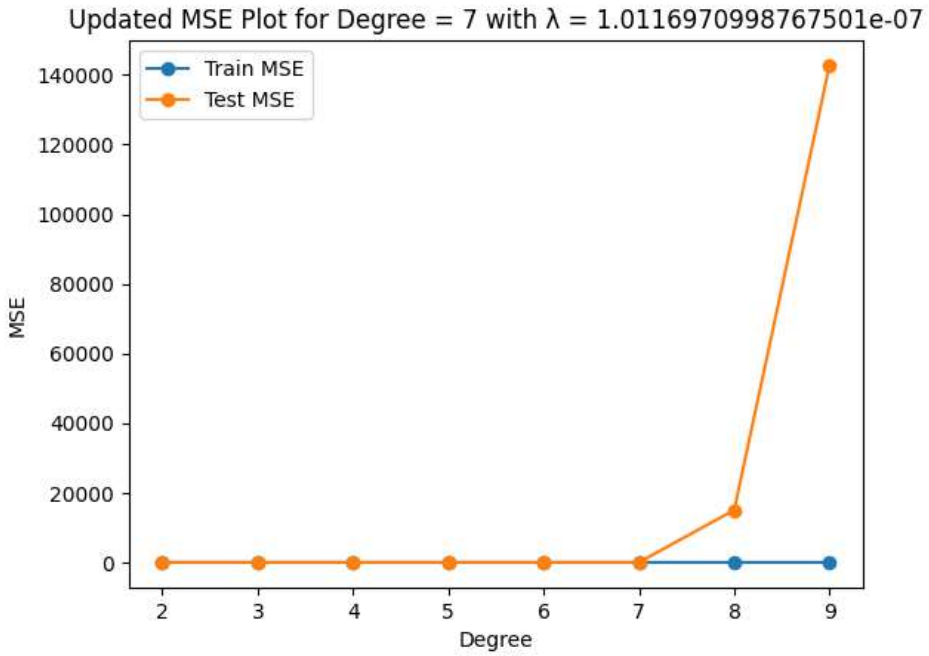


degree 7 for 10




Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 7 with λ = 1.0116970998767501e-07
Actual Output
Predicted Output
Output Test
Input Test
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0


Train MSE
Test MSE
MSE
ln λ
0.000
0.005
0.010
0.015
0.020
0.025
0.030
−17.0
−16.5
−16.0
−15.5
−15.0



Before Regularisation:

w0 value of degree 7: 661.2194233105813

w1 value of degree 7: -16514.17933453462

w2 value of degree 7: 134006.03594925953

w3 value of degree 7: -528069.090907625

w4 value of degree 7: 1147882.1579622647

w5 value of degree 7: -1410531.8200313332

w6 value of degree 7: 920269.6028747887

w7 value of degree 7: -248199.66743487303

MSE for degree 7 training data: 0.00038551116517127477

MSE Percent for degree 7 training data: 0.017790608214033176

MSE for degree 7 test data: 15798.824106349884

MSE Percent for degree 7 test data: 789892.014070655

After Regularization

w0 value of degree 7 3.3746083403289644

w1 value of degree 7 -21.729290990471586

w2 value of degree 7 28.35253310821638

w3 value of degree 7 26.462909222582894

w4 value of degree 7 -10.99251984045241

w5 value of degree 7 -30.86951163586977

w6 value of degree 7 -16.190379844246817

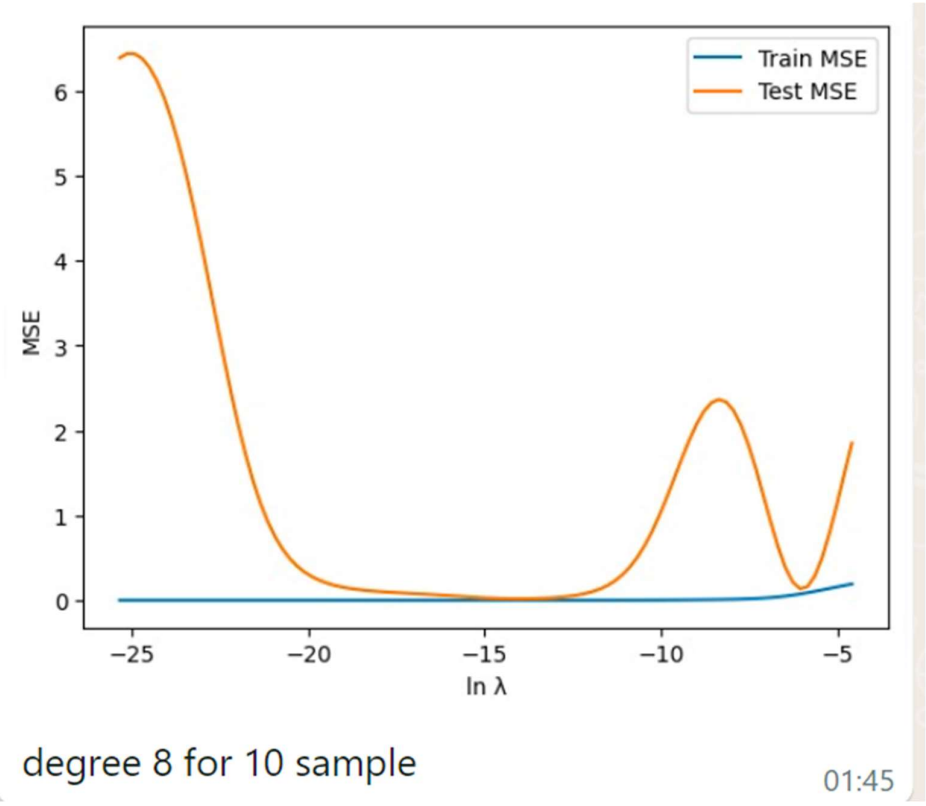
w7 value of degree 7 24.550057674634193

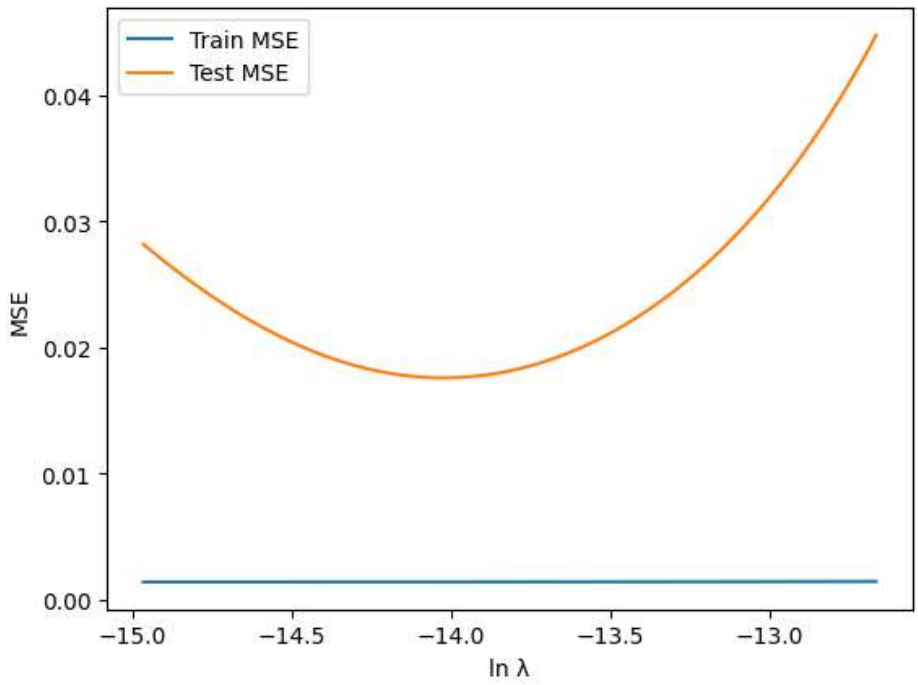
MSE for degree 7 training data: 0.0014130773784549987

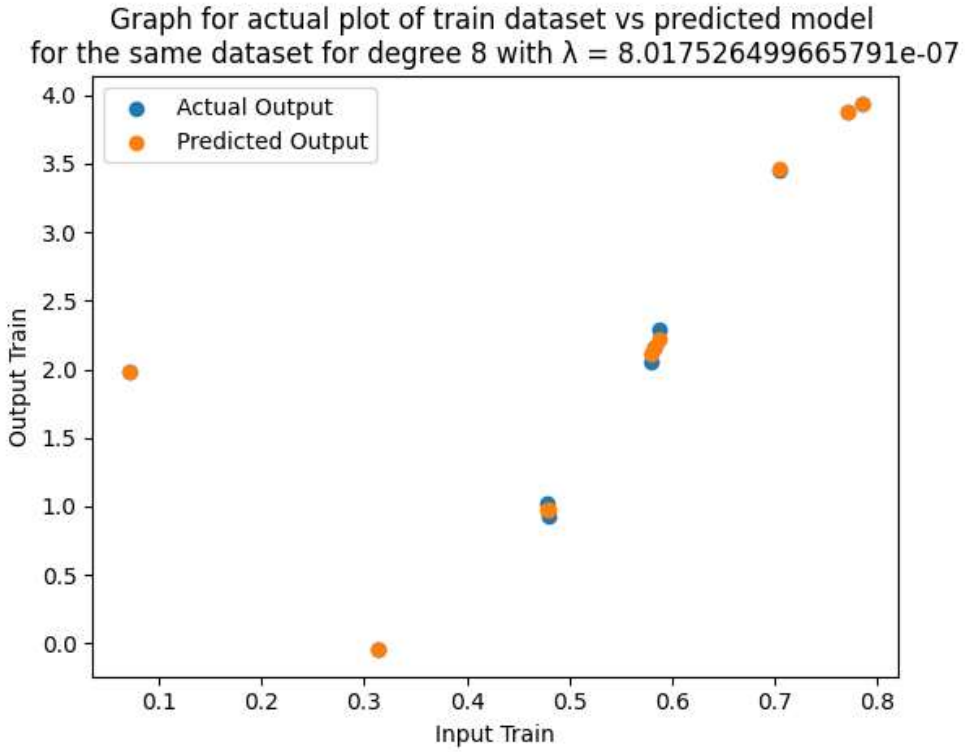
MSE Percent for degree 7 training data: 0.06521083768102279

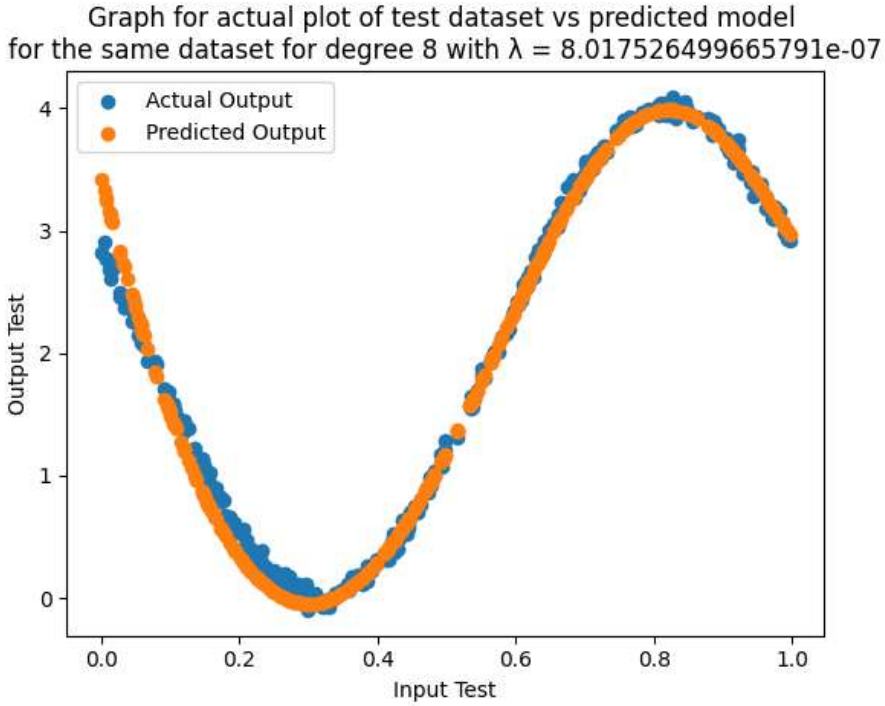
MSE for degree 7 test data: 0.015885713796569716

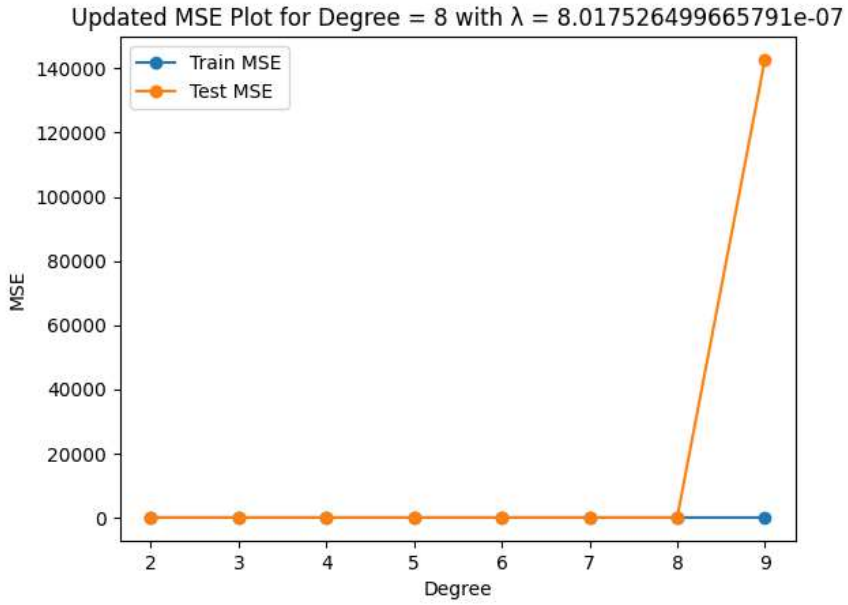
While Regularization






Graph for actual plot of train dataset vs predicted model
for the same dataset for degree 8 with λ = 8.017526499665791e-07
Actual Output
Predicted Output
Output Train
Input Train
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 8 with λ = 8.017526499665791e-07
Actual Output
Predicted Output
Output Test
Input Test
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0



Before Regularisation:

w0 value of degree 8: 594.2141609971521

w1 value of degree 8: -14731.443545146203

w2 value of degree 8: 117599.53323687114

w3 value of degree 8: -451171.40324714105

w4 value of degree 8: 938622.3618639546

w5 value of degree 8: -1064768.026673146

w6 value of degree 8: 577191.911177756

w7 value of degree 8: -59788.76334621701

w8 value of degree 8: -44085.941592659074

MSE for degree 8 training data: 0.7307280313531523

MSE Percent for degree 8 training data: 33.72171104574884

MSE for degree 8 test data: 15016.05242810468

MSE Percent for degree 8 test data: 750755.8673976844

After Regularisation

w0 value of degree 8 3.4233078078963617

w1 value of degree 8 -22.786692300427877

w2 value of degree 8 34.256135753792464

w3 value of degree 8 15.160497842321675

w4 value of degree 8 -9.308558409678081

w5 value of degree 8 -18.02841359821405

w6 value of degree 8 -12.821574732042158

w7 value of degree 8 -0.43748357738889426

w8 value of degree 8 13.48128553573072

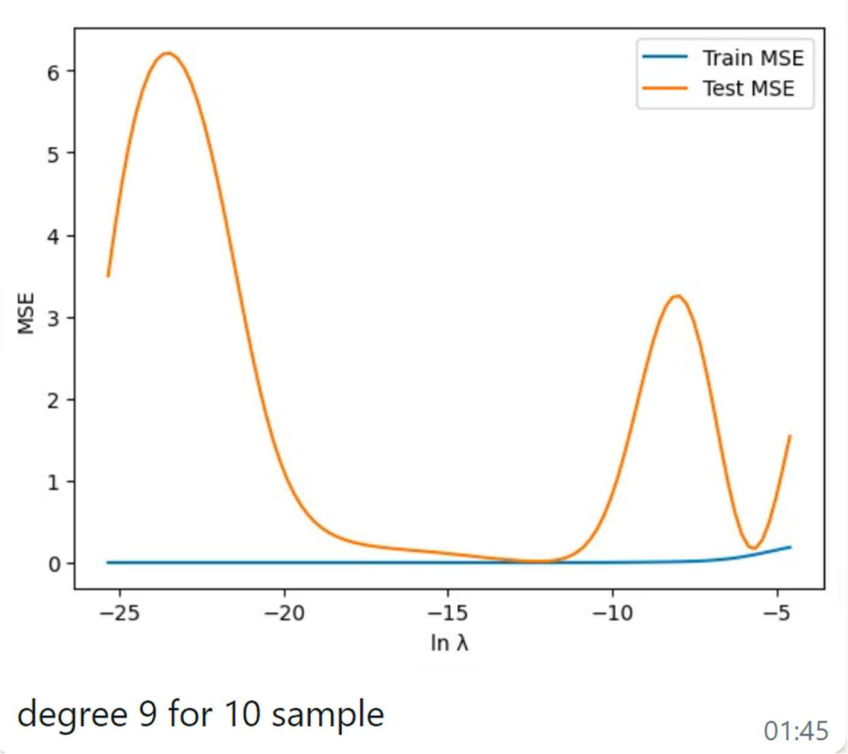
MSE for degree 8 training data: 0.0014182137773695383

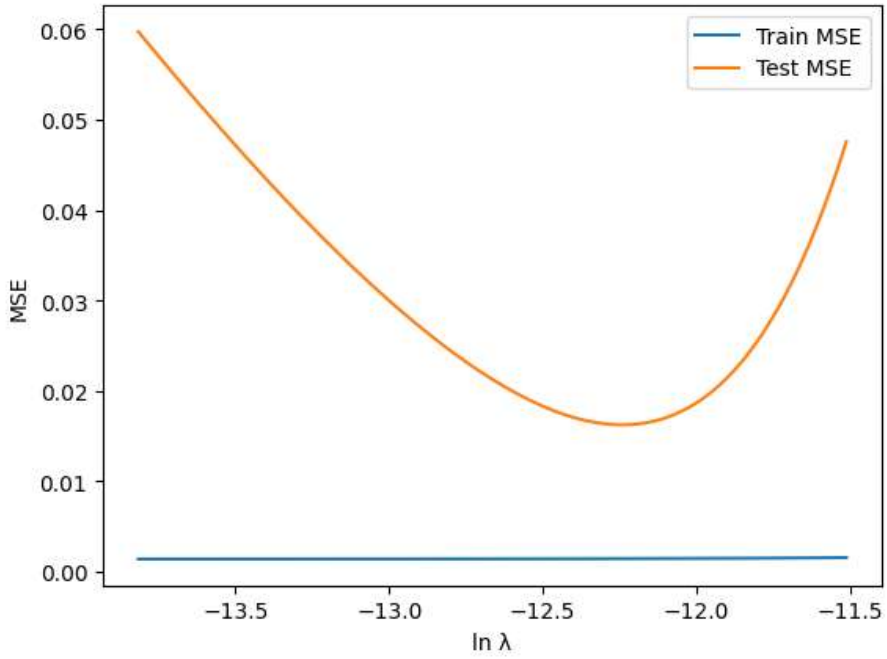
MSE Percent for degree 8 training data: 0.06544787273726807

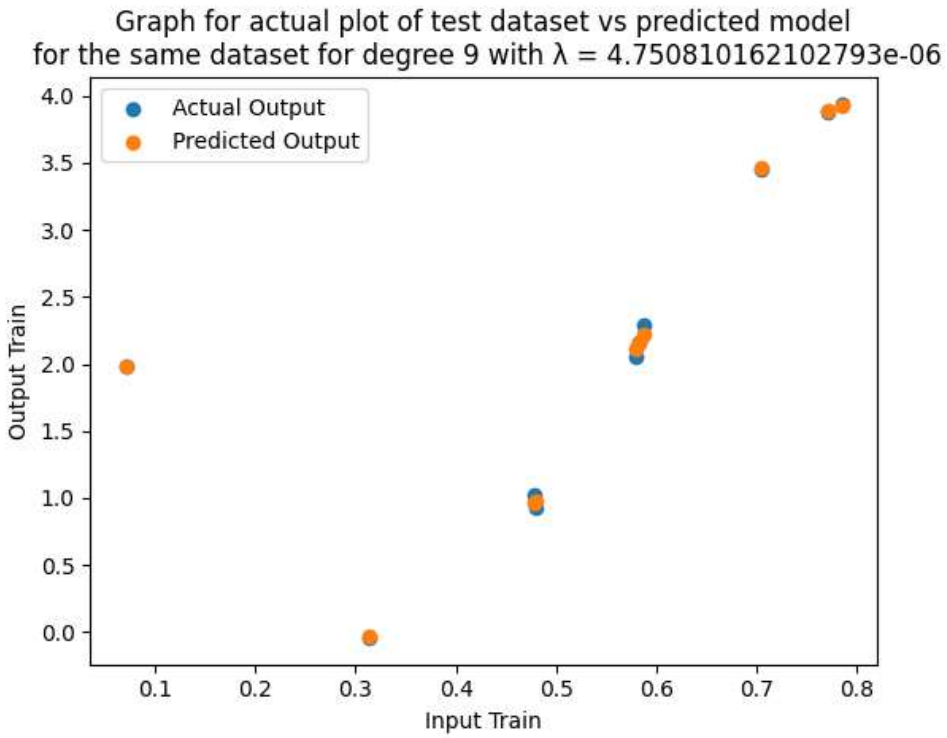
MSE for degree 8 test data: 0.017589570659159497

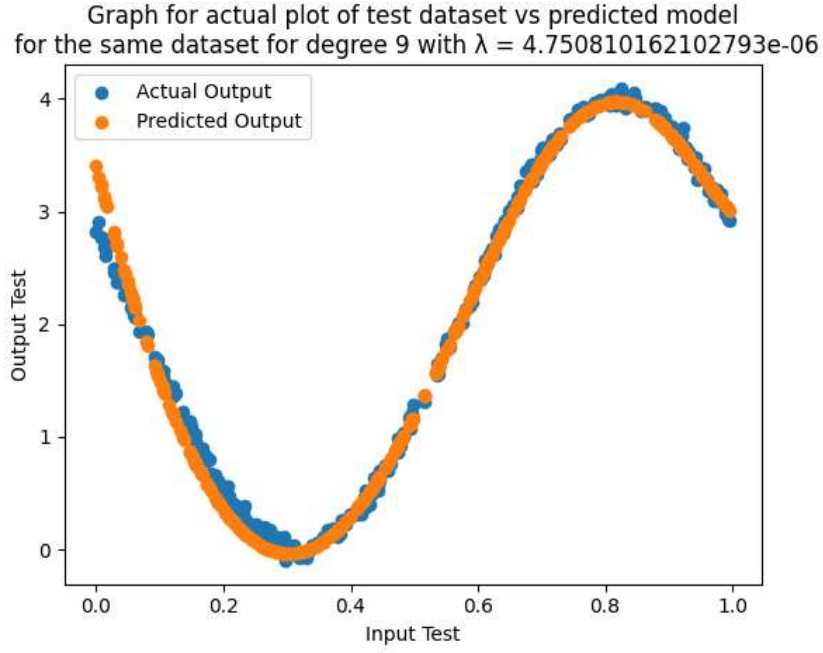
MSE Percent for degree 8 test data: 0.8794237660394835

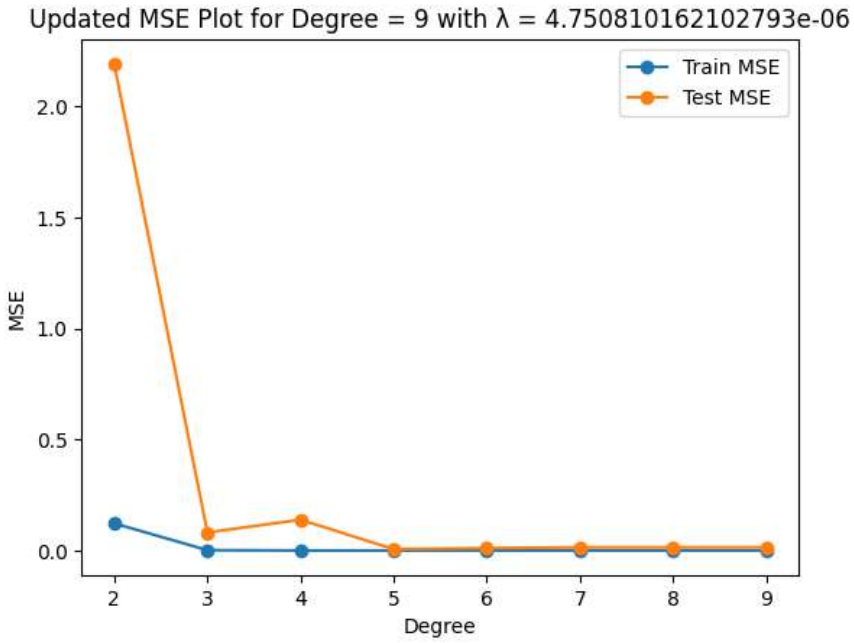
While Regularization






Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 4.750810162102793e-06
Actual Output
Predicted Output
Output Train
Input Train
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Graph for actual plot of test dataset vs predicted model
for the same dataset for degree 9 with λ = 4.750810162102793e-06
Actual Output
Predicted Output
Output Test
Input Test
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0



Before Regularisation:

w0 value of degree 9: 1926.5237413047523

w1 value of degree 9: -47658.99585474458

w2 value of degree 9: 372791.2001995645

w3 value of degree 9: -1309652.1653685276

w4 value of degree 9: 1934633.8951865165

w5 value of degree 9: 674701.8625535371

w6 value of degree 9: -6805151.5890938165

w7 value of degree 9: 10242084.396416912

w8 value of degree 9: -6852725.395698675

w9 value of degree 9: 1790775.0638955447

MSE for degree 9 training data: 2.2338467838849048

MSE Percent for degree 9 training data: 103.0877871581697

MSE for degree 9 test data: 142760.72260319252

MSE Percent for degree 9 test data: 7137591.630119797

After Regularisation

w0 value of degree 9 3.4013004753377016

w1 value of degree 9 -22.50930547417503

w2 value of degree 9 34.11803955960956

w3 value of degree 9 13.095270924339292

w4 value of degree 9 -8.003077413947027

w5 value of degree 9 -14.446381304626874

w6 value of degree 9 -11.075909720792755

w7 value of degree 9 -3.9338991767420315

w8 value of degree 9 3.3089008273782436

w9 value of degree 9 9.033992455921007

MSE for degree 9 training data: 0.001449419484739721

MSE Percent for degree 9 training data: 0.06688795687495586

MSE for degree 9 test data: 0.016259295057436722

MSE Percent for degree 9 test data: 0.8129141279018246

2) Model:

Linear model for regression using Gaussian basis functions with the centres of clusters as the centres of Gaussian basis functions for Datasets 2. Clusters may be formed using the K-means clustering method for the training data.

Gaussian Basis Function:

Gaussian basis functions, also known as radial basis functions (RBFs), are mathematical functions that take the form of a Gaussian probability distribution. These functions are characterized by a peak at a specific location and a bell-shaped curve that decreases as you move away from the peak. Gaussian basis functions are used in a variety of applications, including machine learning, signal processing, interpolation, and function approximation.

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

Here, the components of the function are as follows:

- x is the input variable.
- μ represents the centre of the Gaussian function, indicating the location of the peak.
- σ is the standard deviation of the Gaussian, controlling the width of the bell curve.

Dataset description:

-> 2-Dimensional (Bivariate) input data.
-> Total data = 10200
-> Train-test split = 70:30

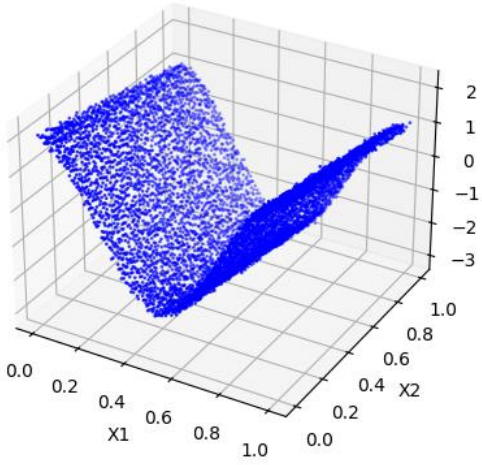


Fig 1: 3-D plot for Train data

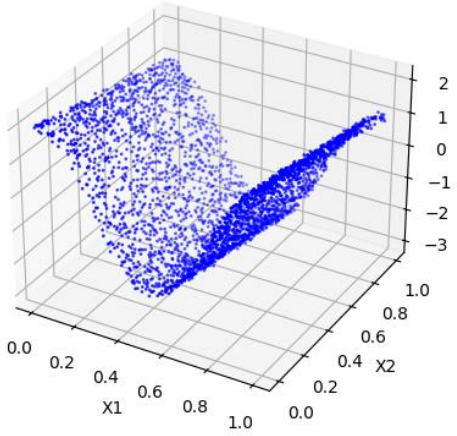
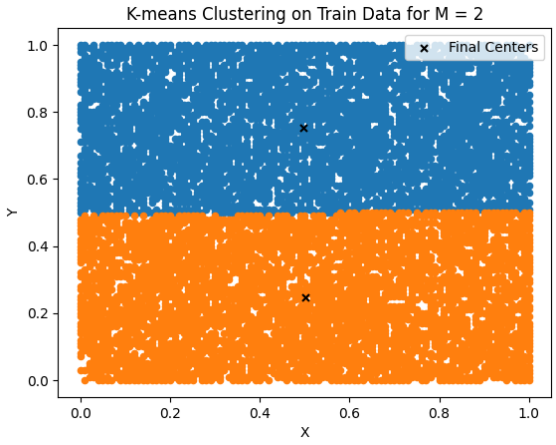
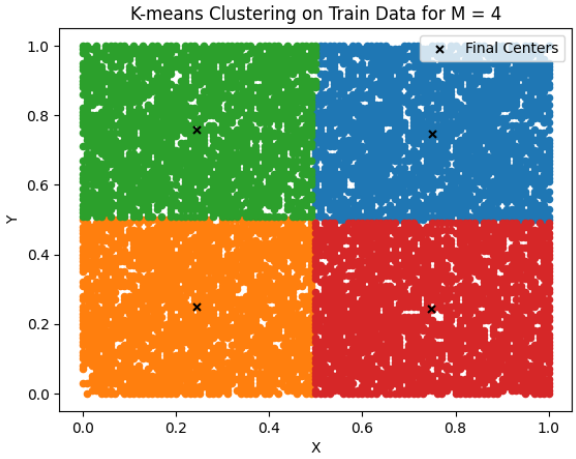
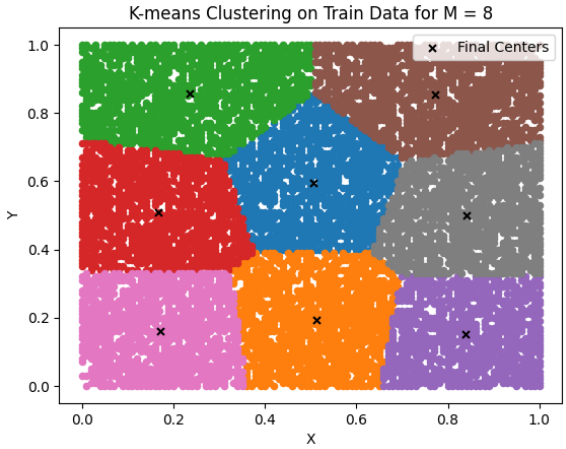
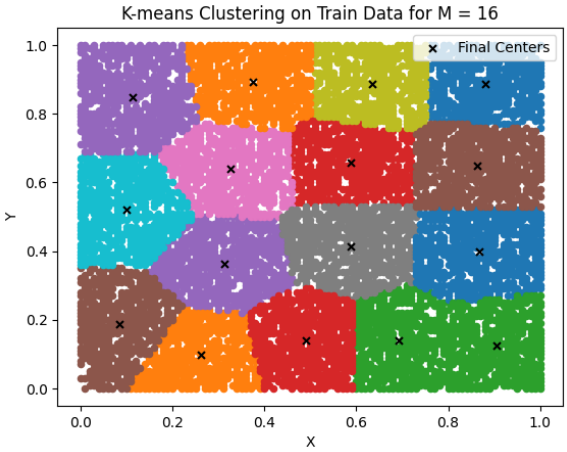


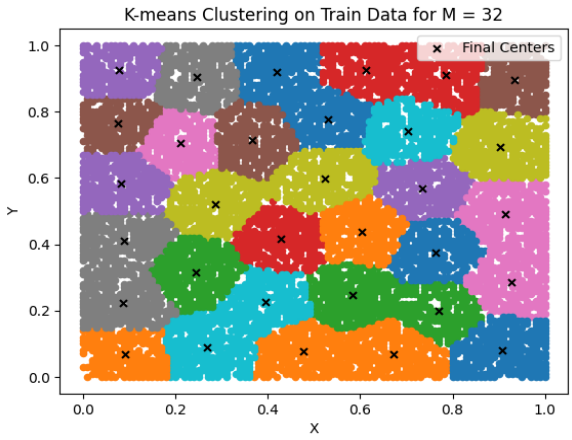
Fig 2: 3-D plot for Test data

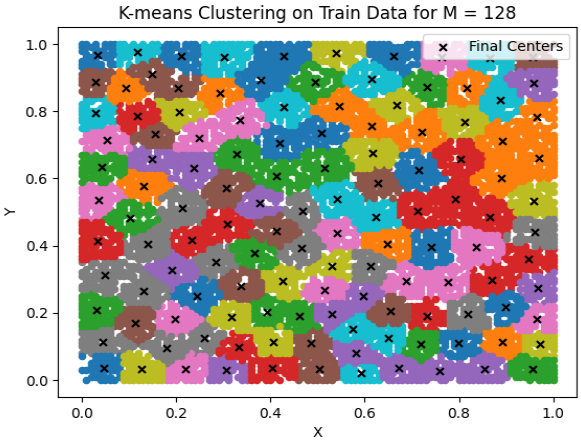

K-means Clustering on Train Data for M = 2
Final Centers
X
Y


K-means Clustering on Train Data for M = 4
Final Centers
X
Y


K-means Clustering on Train Data for M = 8
Final Centers
X
Y


K-means Clustering on Train Data for M = 16
Final Centers
X
Y


K-means Clustering on Train Data for M = 32
Final Centers
X
Y


K-means Clustering on Train Data for M = 128
Final Centers
X
Y

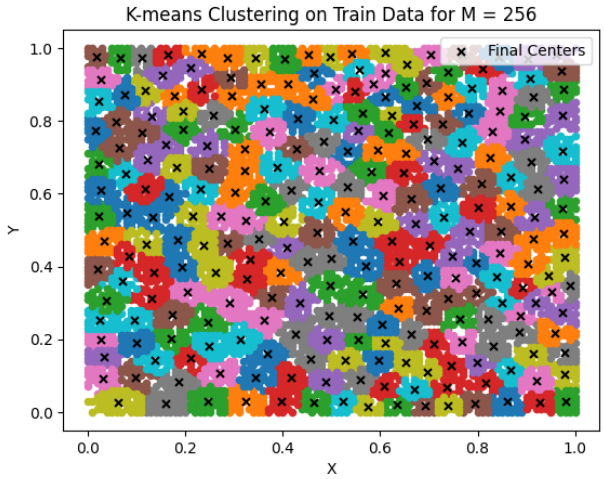
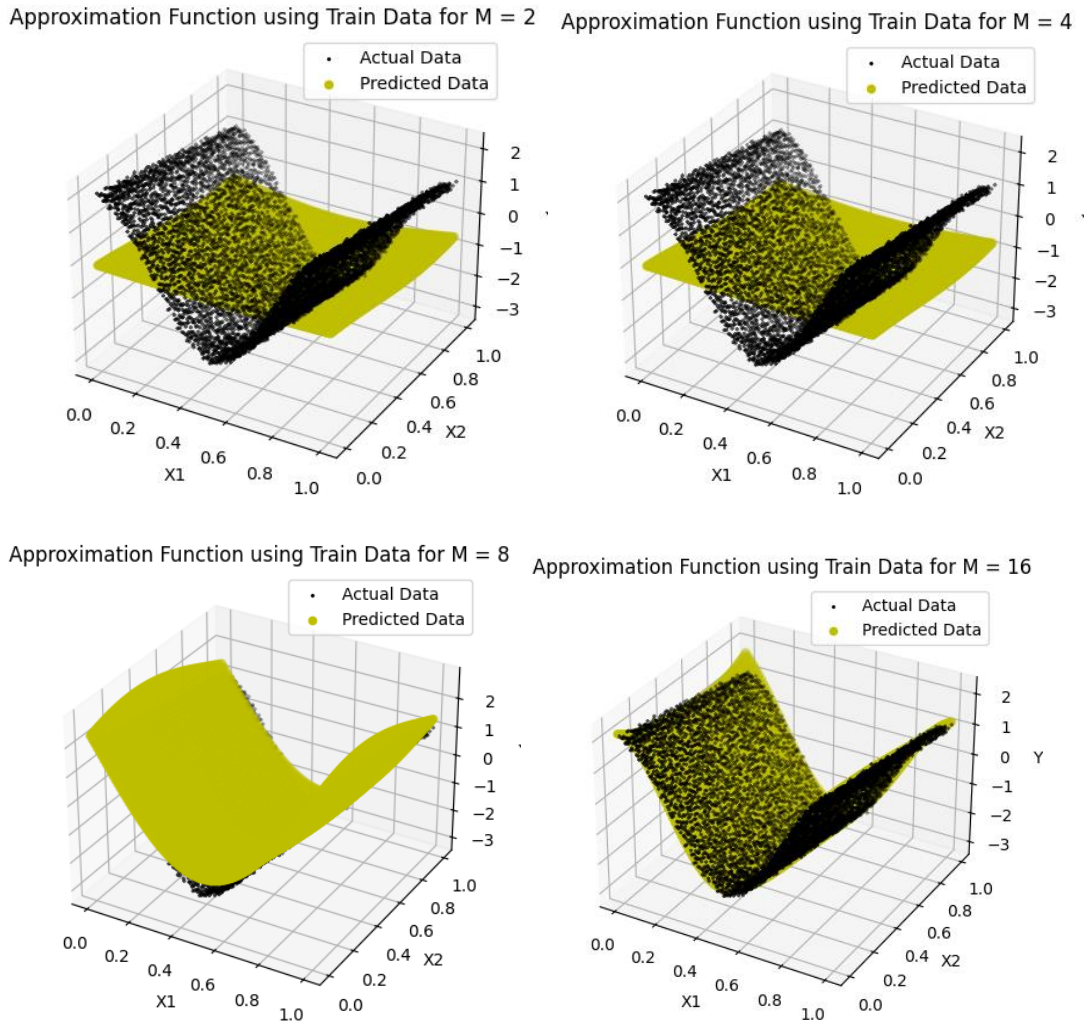


Fig 4: K-means centres for different value of M (2,4,8,16,32,128 and 256)

1) Surface fitting using mesh for actual Train data with different complexities M:



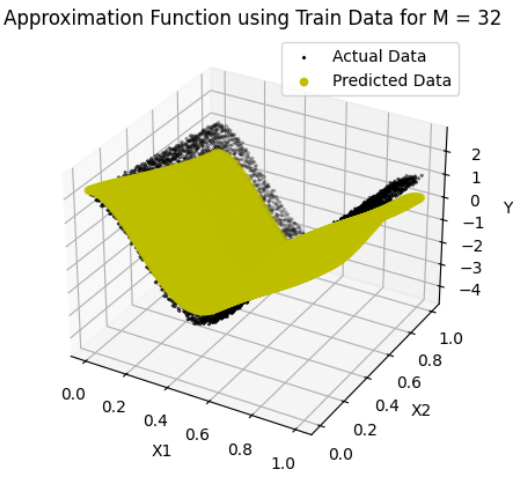


Fig 5: Surface fitting plot for different complexities

2) Mesh plot before and after regularization for M =128 and 256:

For M = 128:

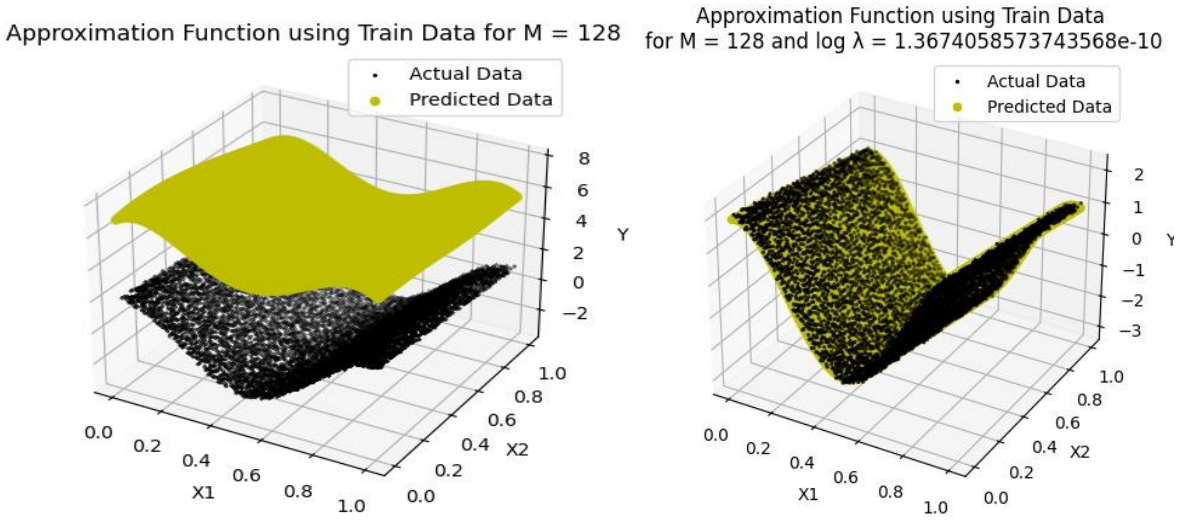


Fig 6: Surface fitting Mesh plot for M=128

For M = 256:

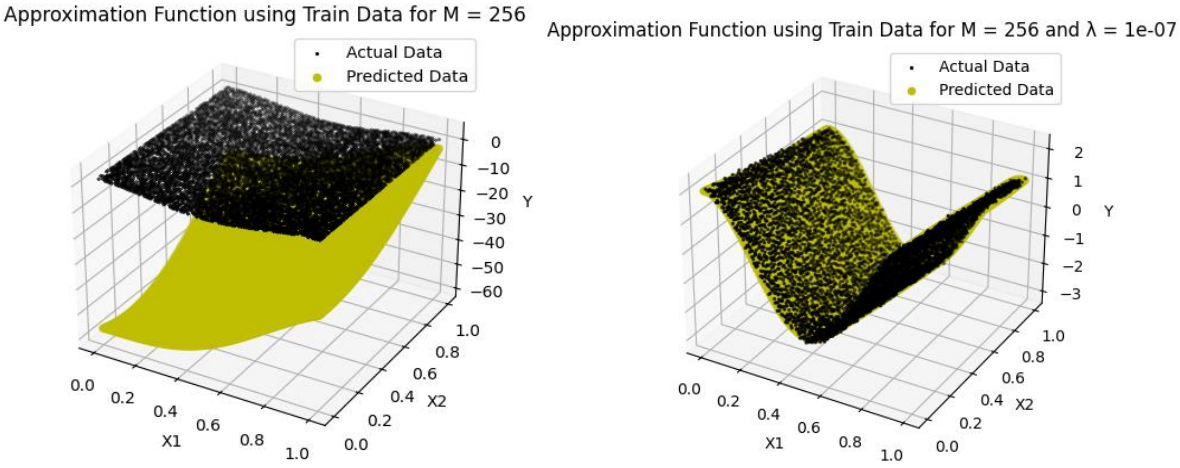


Fig 7: Surface fitting Mesh plot for M=256

3) Plots of the values of mean squared error (MSE) on training data, and test data, for different mo del complexities and for different values of regularization parameter.

a) Without Regularization for Train data:

Values of M (model compl exity)	Mean Square Error (MSE)
2	1.8124232526973425
4	1.9040404973776073
8	0.11120330265563967
16	0.01017353797702641
32	0.49973298539341343
128	40.311695755213705
256	1602.6721182283602

b) Without Regularization for Test data:

Values of M (model compl exity)	Mean Square Error (MSE)
2	1.8163081735983713
4	1.9109007725761011
8	0.11073450698744013
16	0.010500679950180034
32	0.5156279950770379
128	40.08196275309508
256	1562.5996869644225

Regularization plot:

1) Before Regularization plot for MSE vs. Complexities:

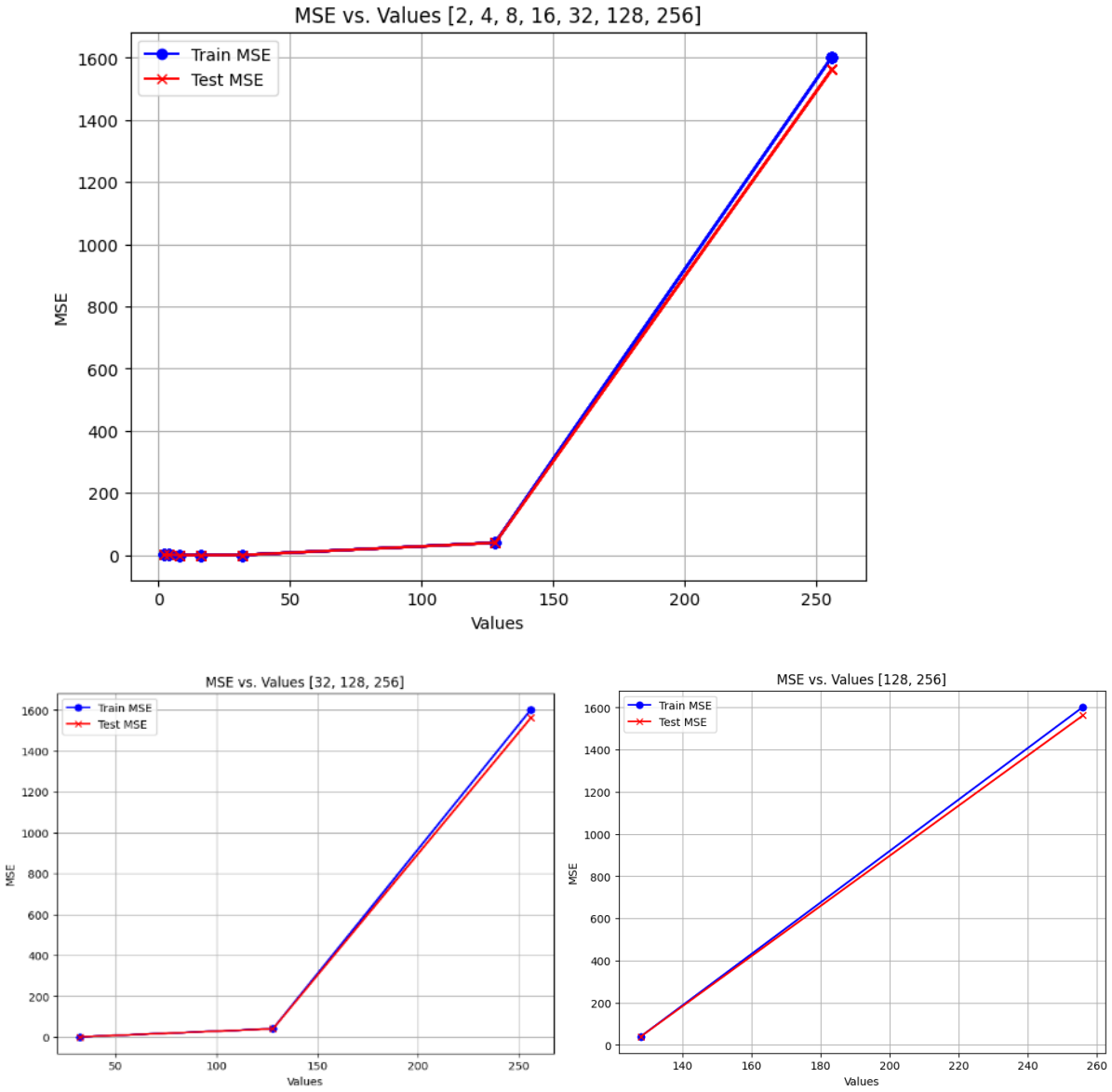


Fig 8: Complexities vs MSE plot for Test and Train for different combinations of M values before Regularization.

2) After Regularization:

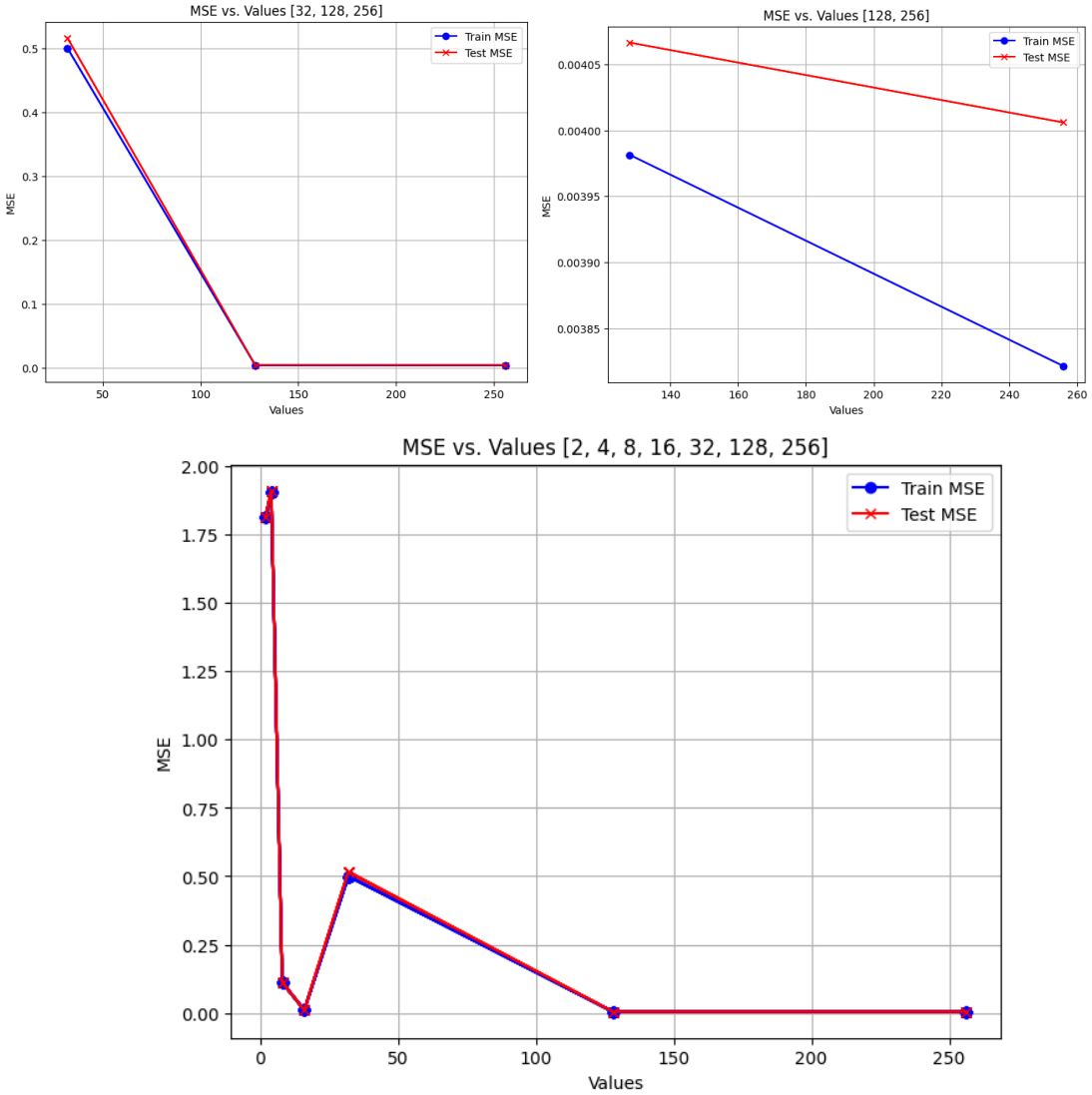
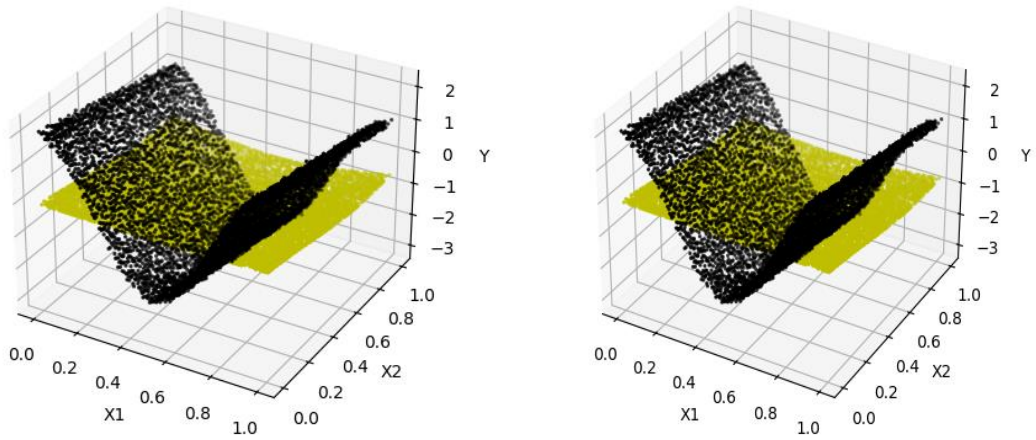


Fig 9: Complexities vs MSE plot for Test and Train for different combinations of M values after regularization.

4) Plots of model output and target output for training data, and test data.

a) Plot of model output and target output for Training data:

Model Output and Target Output for Train Data at M = 2 Model Output and Target Output for Train Data at M = 4



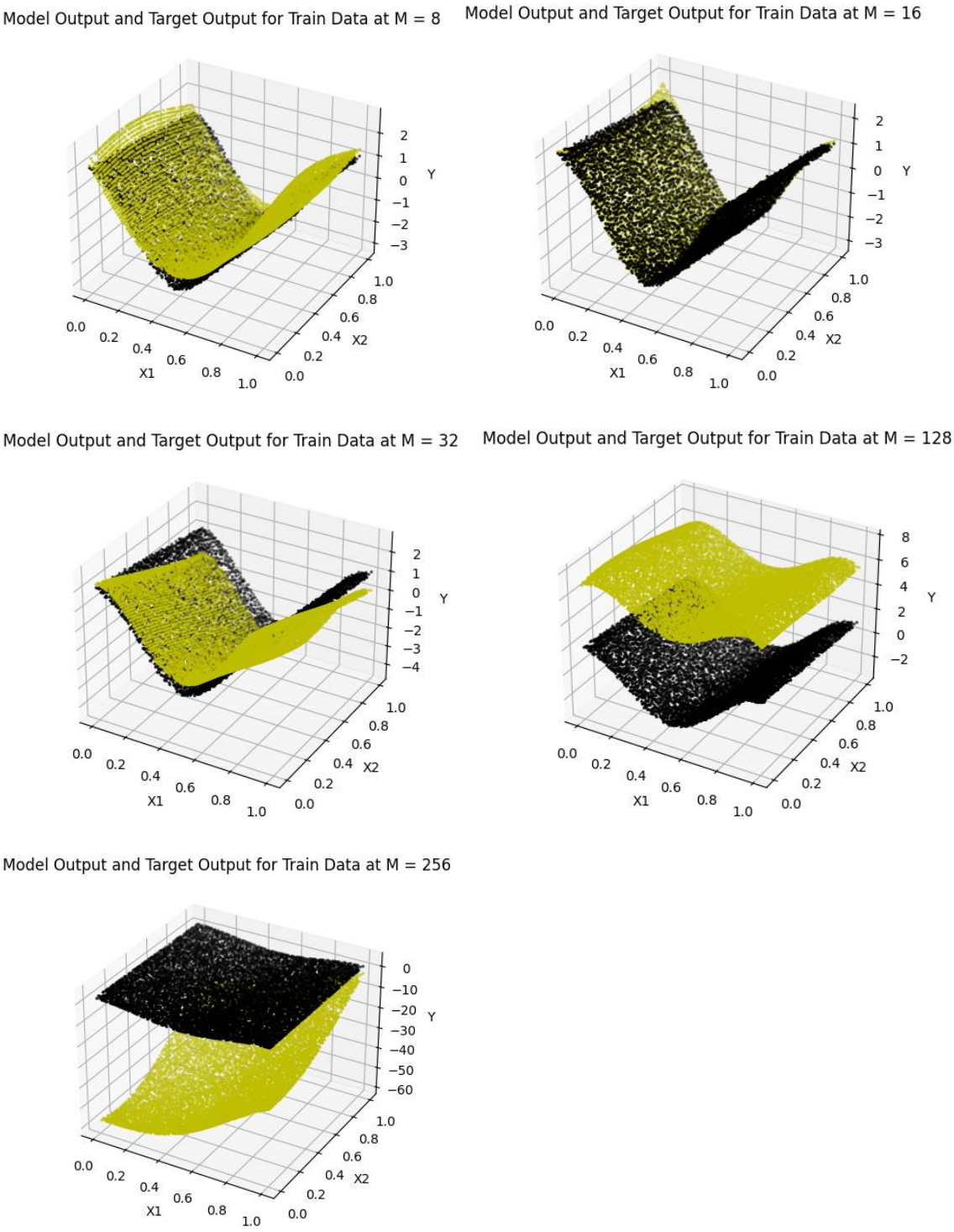
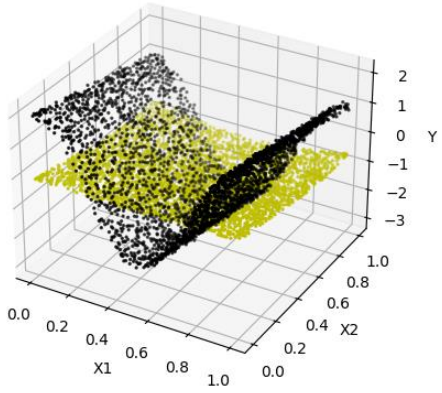


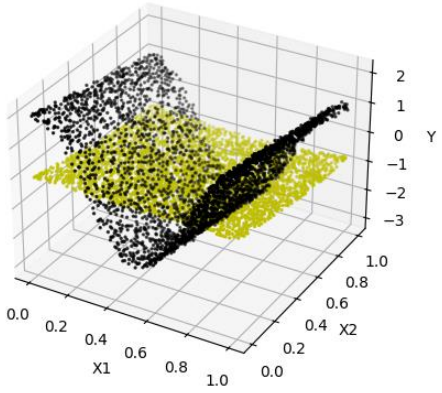
Fig 10: Plot of model output and target output of Training data for different Complexities.

b) Plot of model output and target output for Test data:

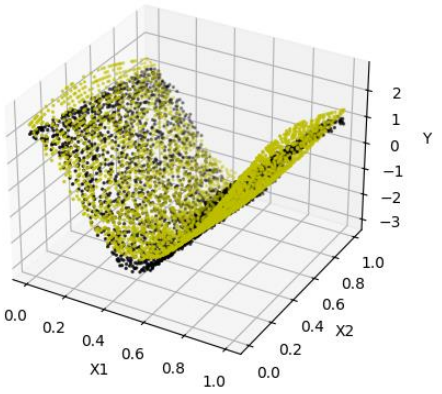
Model Output and Target Output for Test Data at M = 2



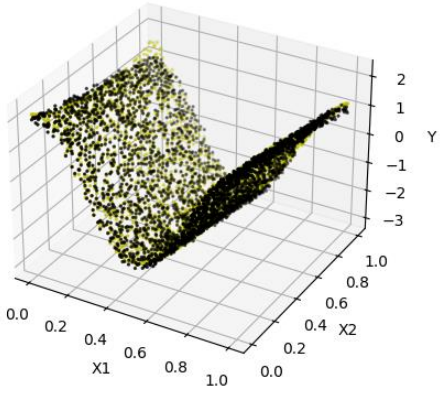
Model Output and Target Output for Test Data at M = 4



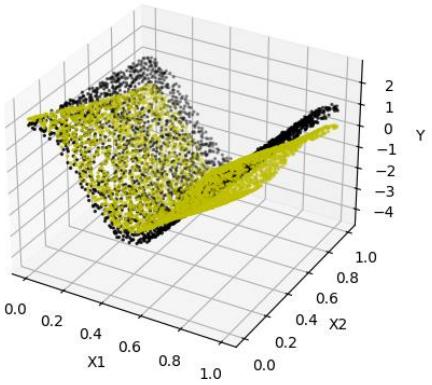
Model Output and Target Output for Test Data at M = 8



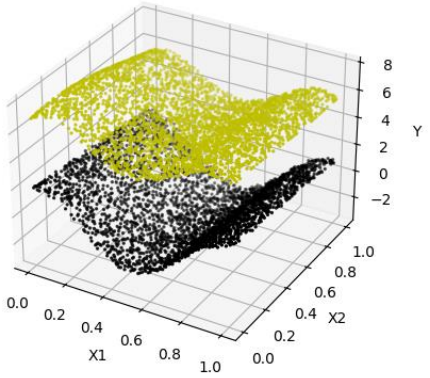
Model Output and Target Output for Test Data at M = 16



Model Output and Target Output for Test Data at M = 32



Model Output and Target Output for Test Data at M = 128



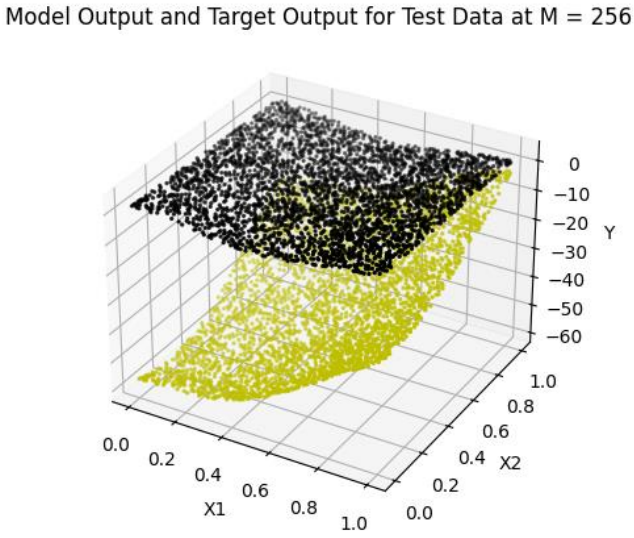
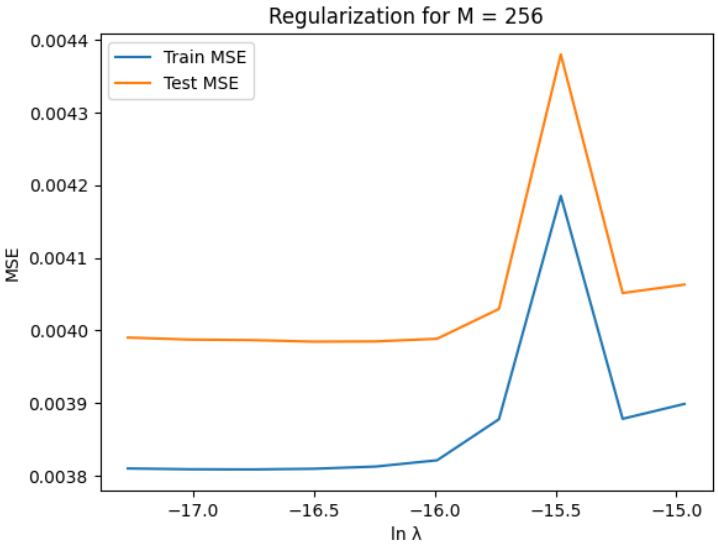


Fig 11: Plot of model output and target output of Test data for different Complexities.

5) Give the plots of model output and target output for training data, and test data for the best regularization parameter on model complexity equal to 256 (For Dataset 2).



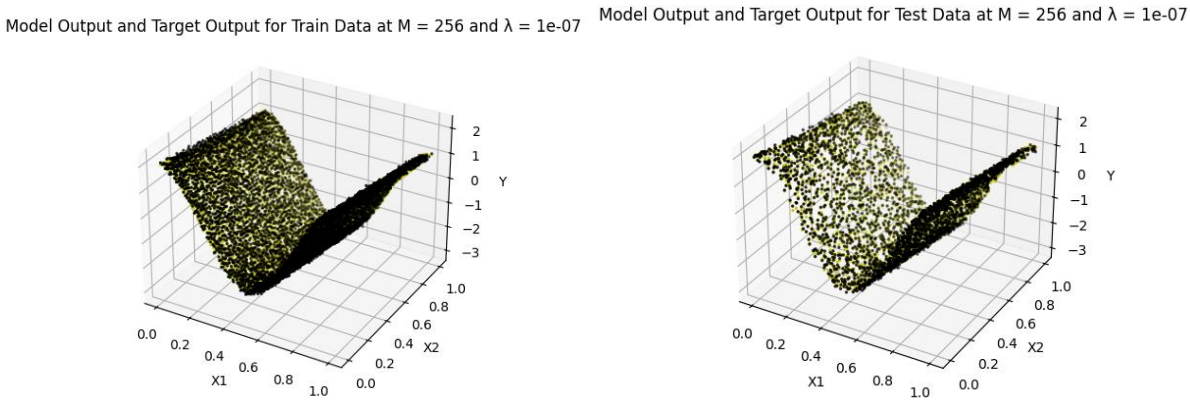


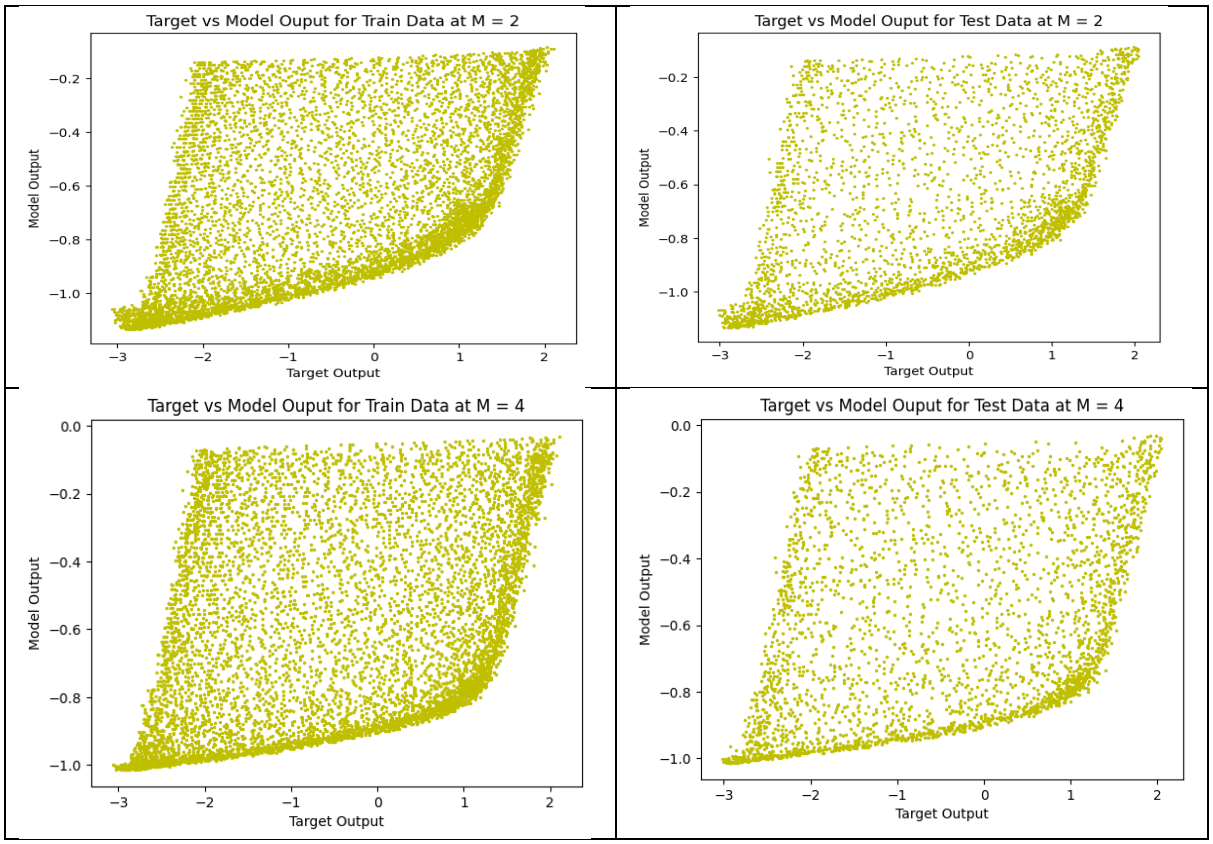
Fig 12: Plot for Model output and Target output for train data and test data after regularization for M = 256

->Regularization parameter value 1e-07.

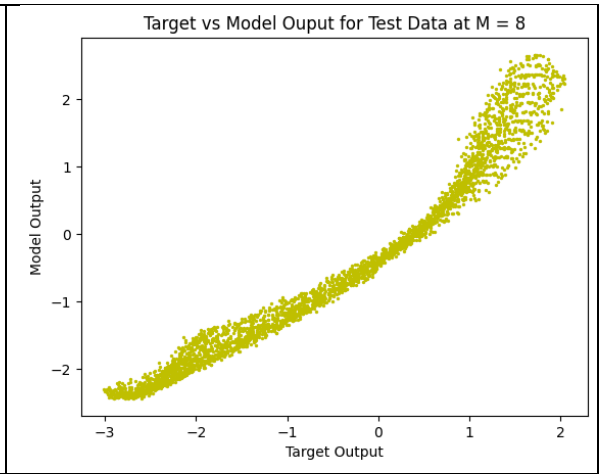
- MSE value before Regularization on Train Data = 1602.6721182283602
- MSE value before Regularization on Test Data = 1562.5996869644225

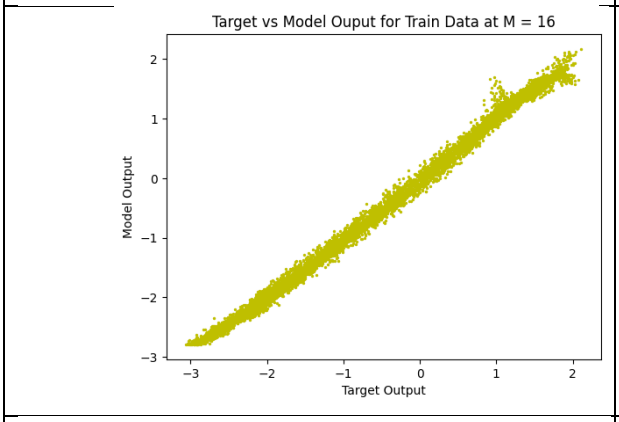
- MSE value after Regularization on Train Data = 0.0038212736788276885
- MSE value after Regularization on Test Data = 0.004006038406441563

6. Scatter plot with target output on x-axis and model output on y-axis, for training data, and test data.



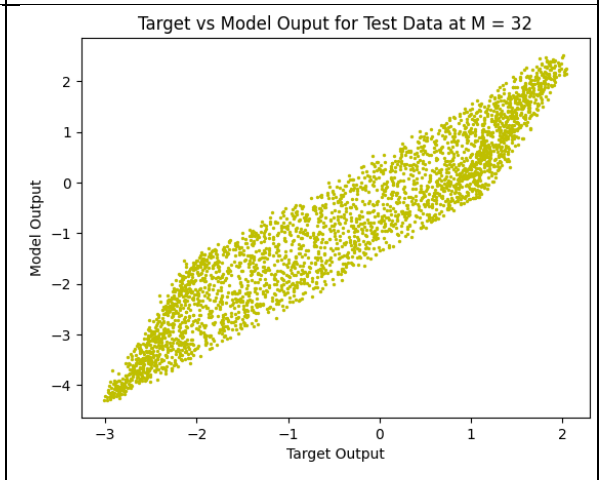

Target vs Model Ouput for Train Data at M = 8
Model Output
Target Output

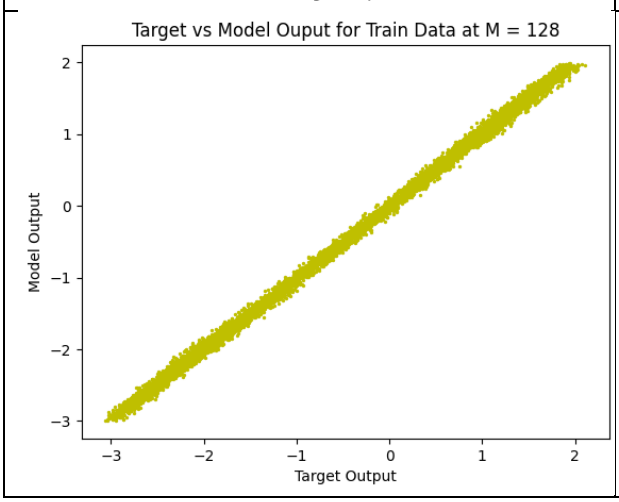

Target vs Model Ouput for Test Data at M = 8
Model Output
Target Output

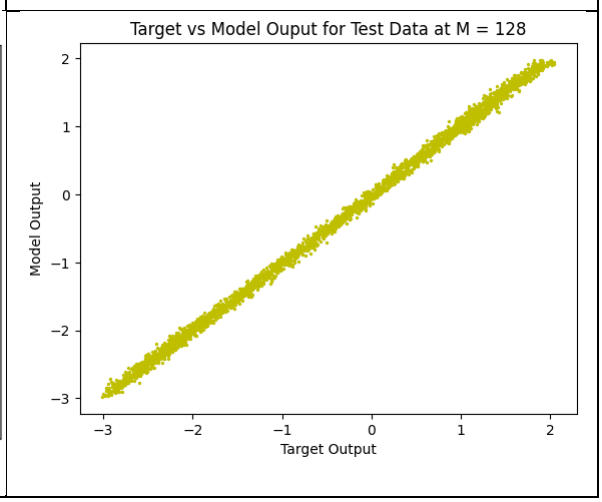

Target vs Model Ouput for Train Data at M = 16
Model Output
Target Output


Target vs Model Ouput for Test Data at M = 16
Model Output
Target Output


Target vs Model Ouput for Train Data at M = 32
Model Output
Target Output


Target vs Model Ouput for Test Data at M = 32
Model Output
Target Output


Target vs Model Ouput for Train Data at M = 128
Model Output
Target Output


Target vs Model Ouput for Test Data at M = 128
Model Output
Target Output

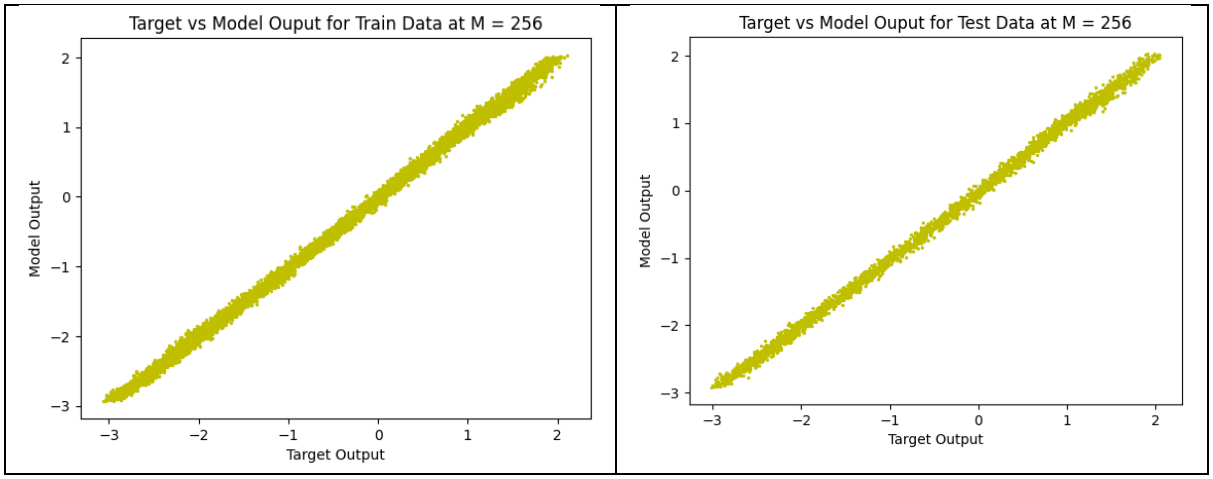


Fig 13: Scatter plot with target output on x-axis and model output on y-axis, for training data, and test data for different value of complexities M.