Function approximation or Regression

Reetwik Das

July 20, 2021

Contents

List of Tables

List of Figures

1 Polynomial curve fitting for dataset

This is majorly used for funstion approximation or regression. We have an independent variable, say x and a target variable or dependent variable, say y and we need to approximate the underlying function $y = f(x)$.
We approximate the function $f(x) = w_0 + w_1x_1 + w_2x_2.....w_Mx_M$, here M is the degree and ususally a higher M would help us getting a better approximation of the function.

Dataset1 : function2.csv is the dataset used in this section for all the experiments.

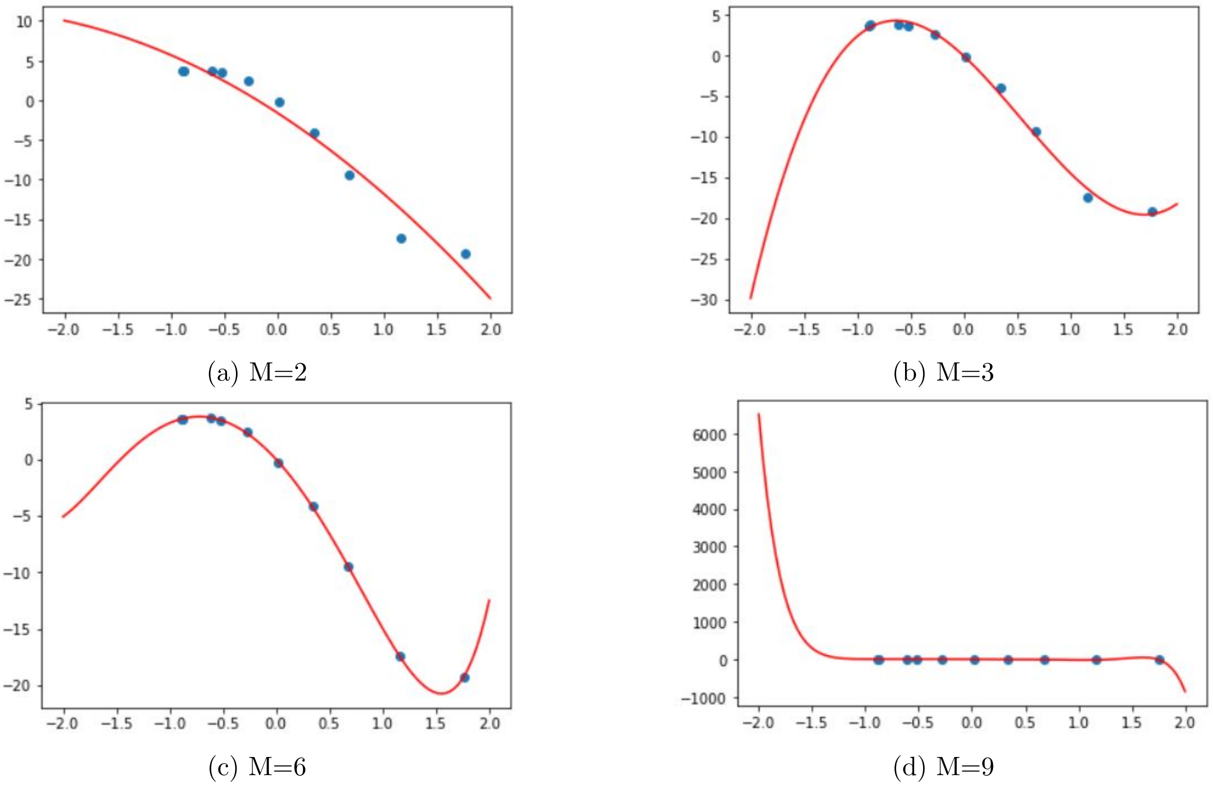
1.1 Without Regularization

1.1.1 Datapoints in training dataset - 10

Hyperparameters $M = \{2, 3, 6, 9\}, \lambda = 0$

M (Hyperparameter)	**Training Dataset**	**Validation Dataset**	**Test Dataset**
2	1.7610	8.9828	11.1838
3	0.5082	1.2438	1.3943
6	0.05811	5.0390	6.3352
9	$8.66 * 10^{-6}$	843.0139	1015.5930

Table 1.1.1: ERMS for dataset1, training size = 10, $\lambda = 0$



(a) M=2

(b) M=3

(c) M=6

(d) M=9

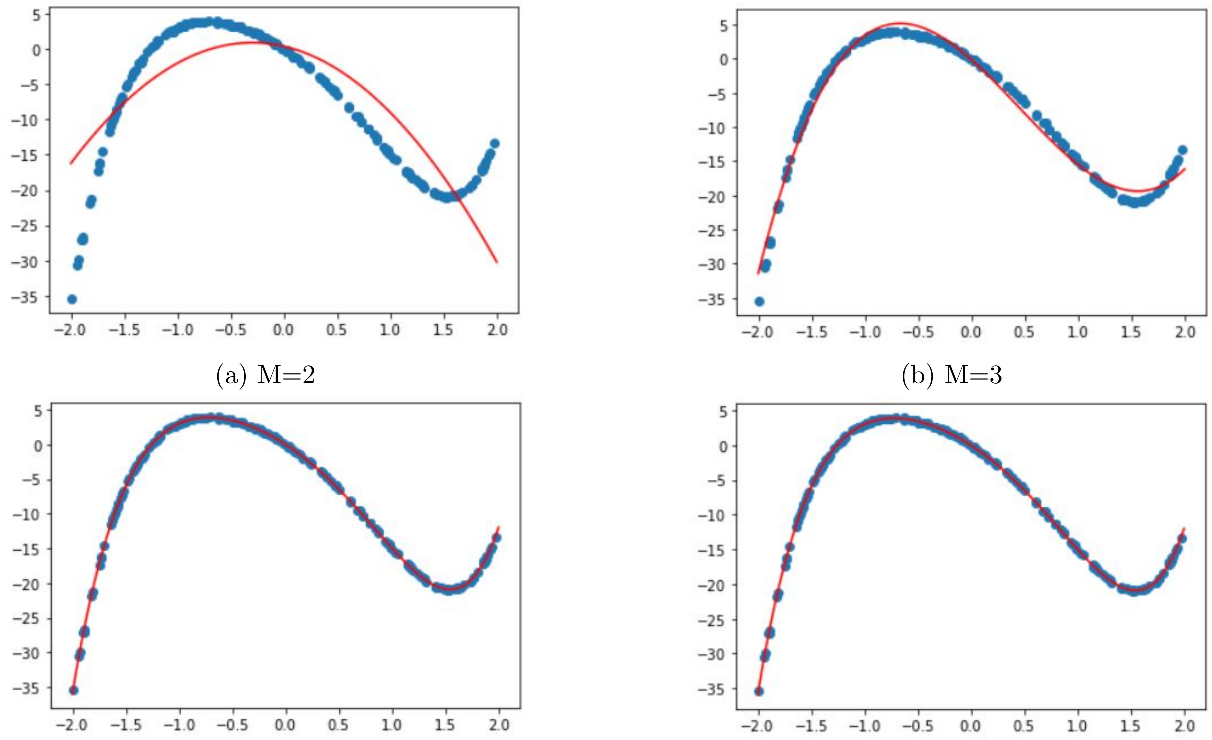
Figure 1.1.1: Approximated function vs Training points for dataset1, training size =10, $\lambda = 0$

1.1.2 Datapoints in training dataset - 200

Hyperparameters $M = \{2, 3, 6, 9\}, \lambda = 0$

M (Hyperparameter)	**Training Dataset**	**Validation Dataset**	**Test Dataset**
2	5.2002	4.7814	5.4460
3	1.1704	1.1519	1.2110
6	0.0923	0.0972	0.1087
9	0.0922	0.0974	0.1089

Table 1.1.2: ERMS for dataset1, training size = 200, $\lambda = 0$



(a) M=2

(b) M=3

(c) M=6

(d) M=9

Figure 1.1.2: Approximated function vs Training points for dataset1, training size = 100, $\lambda = 0$

1.2 With Regularization

1.2.1 Datapoints in training dataset - 10, $\lambda = 0.1$

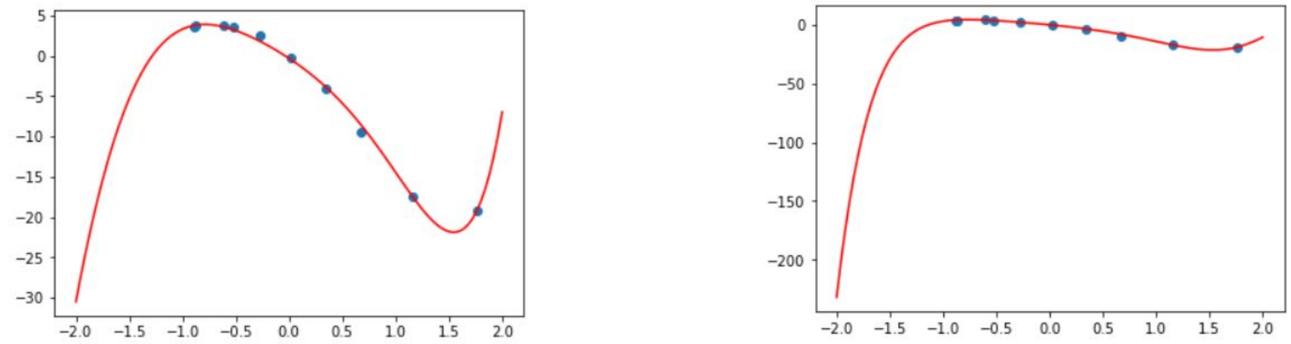
Since we observed overfitting only when dataset size was 10 and M was higher.
Hyperparameters $M = \{6, 9\}, \lambda = 0.1$

M (Hyperparameter)	Training Dataset	Validation Dataset	Test Dataset
6	0.3900	0.9661	1.7154
9	0.4045	28.8458	35.8129

Table 1.2.1: ERMS for dataset1, training size =10, $\lambda = 0.1$

Observations :

We observed in the section that when we don't have enough data for training i.e when we had only 10 points for training and the model complexity is high, the ERMS for training data is pretty low although the ERMS for test and validation is higher. This is because of overfitting and when we have more data i.e 200 points for training and the model complexity is high this problem is resolved.



(a) M=6

(b) M=9

Figure 1.2.1: Approximated function vs Training points for dataset1, training size =10, $\lambda = 0.1$

2 Linear model for regression using Polynomial Basis function

Dataset2 : function2-2d.csv is used for all the experiments in the given section.

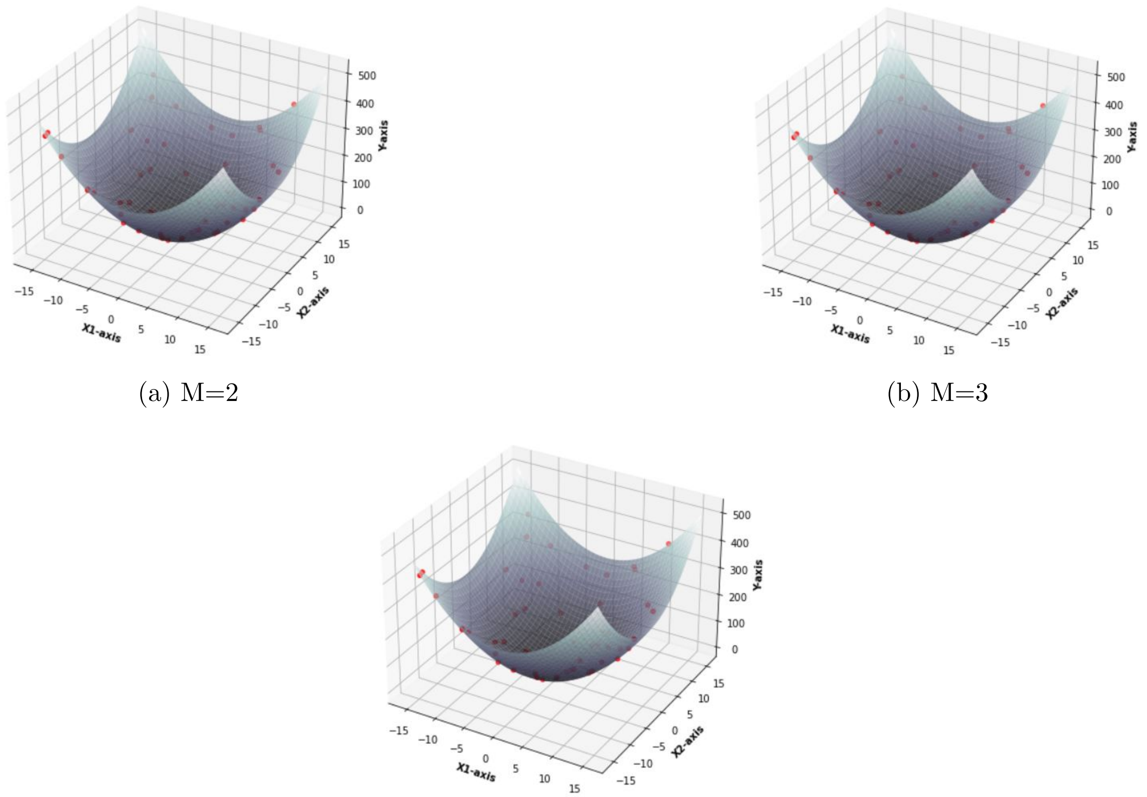
2.1 Without regularization

2.1.1 Datapoints in training dataset - 50

Hyperparameters $M = \{2, 3, 6\}$

M (Hyperparameter)	Training Dataset	Validation Dataset	Test Dataset
2	$1.5604 * 10^{-13}$	$1.5524 * 10^{-13}$	$2.0963 * 10^{-13}$
3	$4.6330 * 10^{-13}$	$5.5056 * 10^{-13}$	$7.7062 * 10^{-13}$
6	$2.1917 * 10^{-10}$	$7.3972 * 10^{-10}$	$2.8797 * 10^{-10}$

Table 2.1.1: ERMS for dataset2, training size = 50, $\lambda = 0$



(a) M=2

(b) M=3

(c) M=6

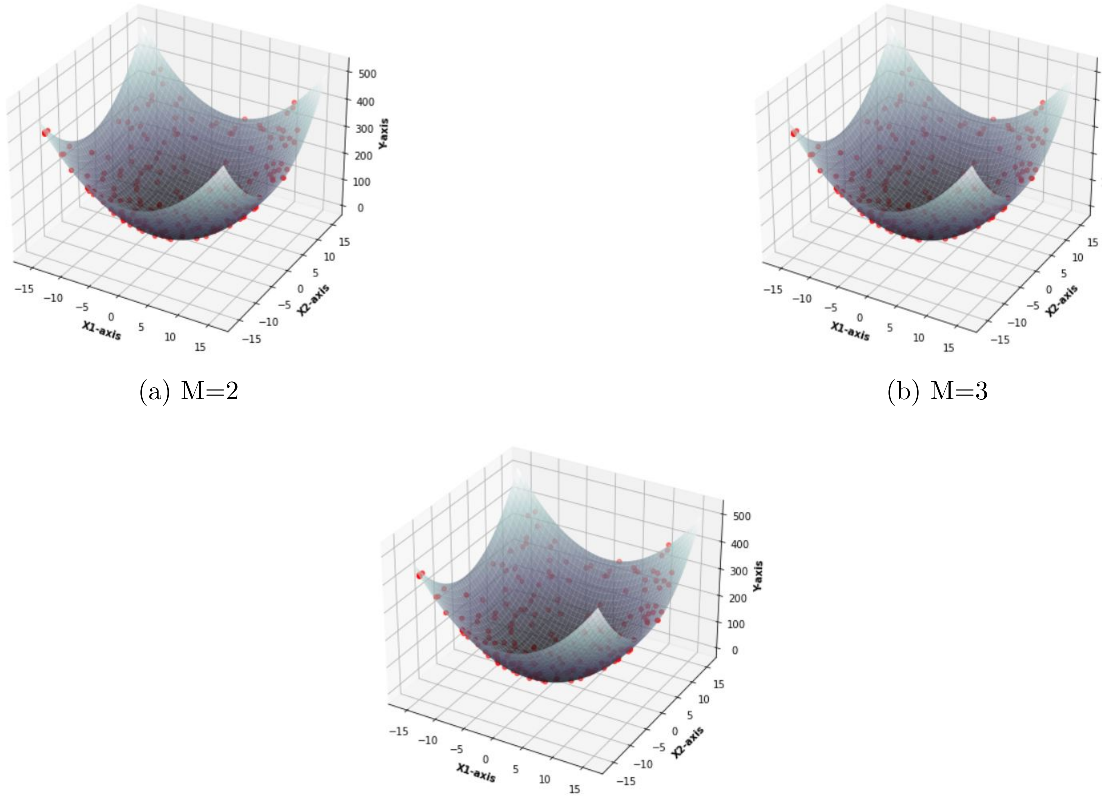
Figure 2.1.1: Approximated surface vs Training points for dataset2, training size = 50, $\lambda = 0$

2.1.2 Datapoints in training dataset - 200

Hyperparameters $M = \{2, 3, 6\}$

M (Hyperparameter)	Training Dataset	Validation Dataset	Test Dataset
2	$1.2609 * 10^{-13}$	$1.1539 * 10^{-13}$	$1.2662 * 10^{-13}$
3	$6.9699 * 10^{-13}$	$7.4280 * 10^{-13}$	$7.2801 * 10^{-13}$
6	$2.2037 * 10^{-10}$	$2.1976 * 10^{-10}$	$2.1677 * 10^{-10}$

Table 2.1.2: ERMS for dataset2, training size = 200, $\lambda = 0$



(a) M=2

(b) M=3

(c) M=6

Figure 2.1.2: Approximated surface vs Training points for dataset2, training size = 200, $\lambda = 0$

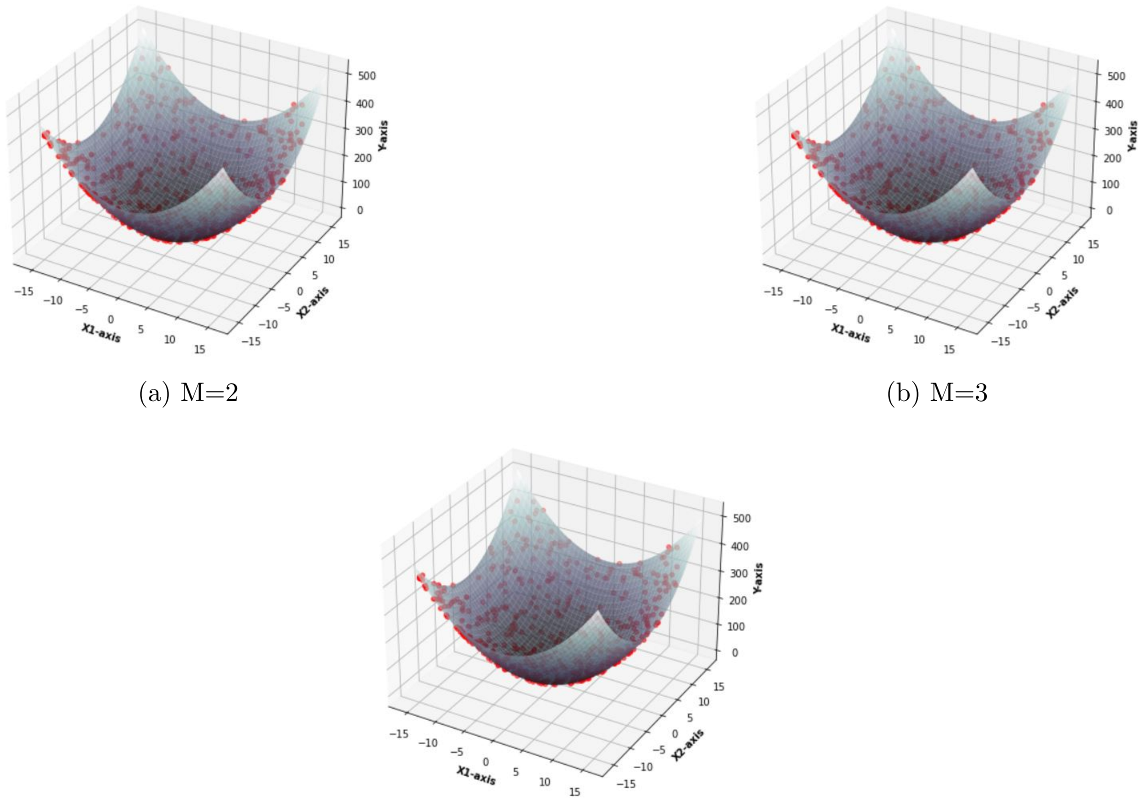
2.1.3 Datapoints in training dataset - 500

Hyperparameters $M = \{2, 3, 6\}$

M (Hyperparameter)	**Training Dataset**	**Validation Dataset**	**Test Dataset**
2	$2.0275 * 10^{-13}$	$2.0586 * 10^{-13}$	$2.0963 * 10^{-13}$
3	$7.9449 * 10^{-13}$	$7.4680 * 10^{-13}$	$7.7062 * 10^{-13}$
6	$3.1754 * 10^{-10}$	$2.8852 * 10^{-10}$	$2.8797 * 10^{-10}$

Table 2.1.3: ERMS for dataset2, training size = 500, $\lambda = 0$

We see that for none of the cases the model complexity is too high for the amount of training data available and hence we don't require and regularization.



(a) M=2

(b) M=3

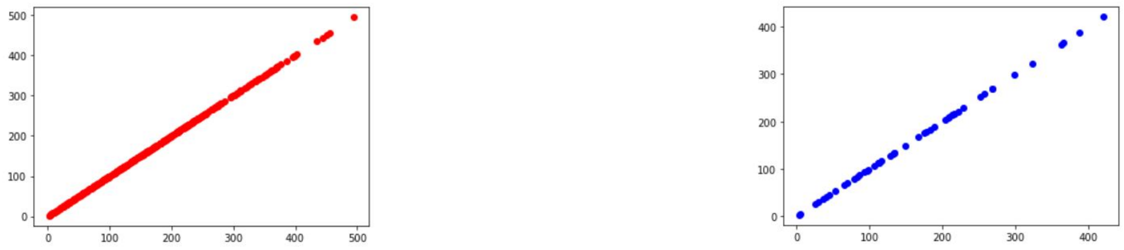
(c) M=6

Figure 2.1.3: Approximated surface vs Training points for dataset2, training size = 500, $\lambda = 0$

2.1.4 Scatter plots, training size = 50

2.1.5 Scatter plots, training size = 200

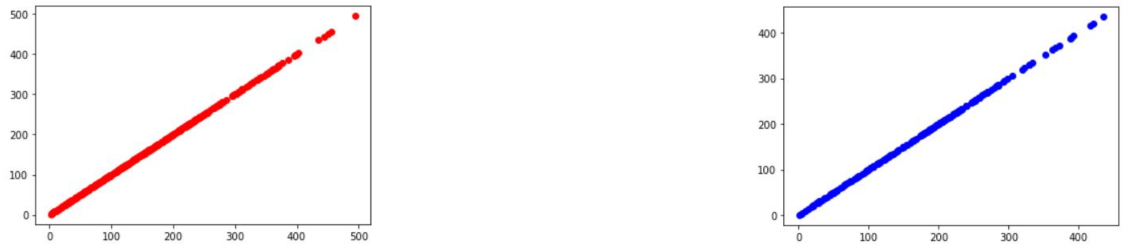
2.1.6 Scatter plots, training size = 500



(a) Testing Data

(b) Training Data

Figure 2.1.4: Model output vs Target output for dataset2, training size = 50, $\lambda = 0$, $M = 2$



(a) Testing Data

(b) Training Data

Figure 2.1.5: Model output vs Target output for dataset2, training size = 200, $\lambda = 0$, $M = 2$

3 Linear model for regression using Gaussian Basis function

For Task 3 the amount of data to be taken for training was not mentioned specifically so we took 70% of the available data as training data.

3.1 Dataset 2 : function2-2d.csv

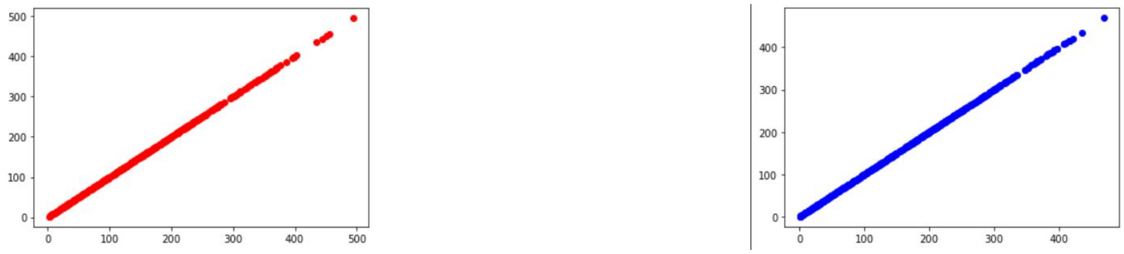
3.1.1 Without regularization

Hyperparameters : $\sigma = 20$, K = Number of clusters = $\{20, 22, 24\}$

K	**Training Dataset**	**Validation Dataset**	**Test Dataset**
20	3.2877	4.0803	4.9529
22	3.0172	3.2468	3.6907
24 (Best)	2.5483	2.9303	3.1383

Table 3.1.1: ERMS for dataset2, $\lambda = 0$, $\sigma = 20$

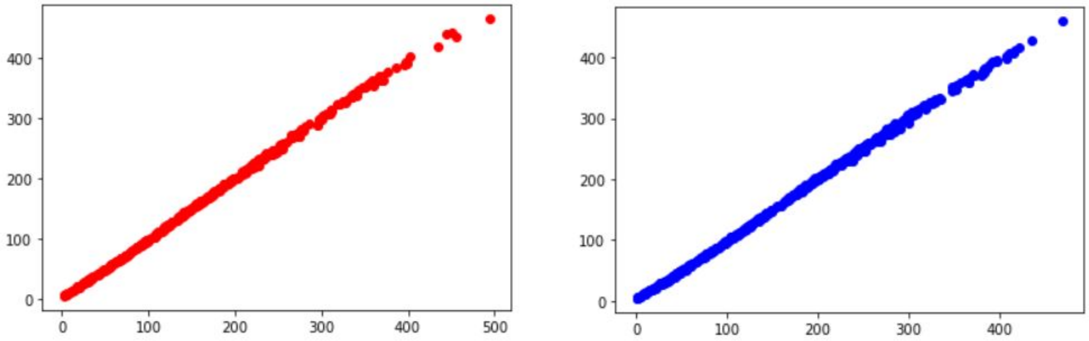
As we can see there is no overfitting observed since we already have enough data even for model with higher complexities like with $K = 24$. Hence no regularization is required.



(a) Testing Data

(b) Training Data

Figure 2.1.6: Model output vs Target output for dataset2, training size = 500, $\lambda = 0$, $M = 3$



(a) Testing Data

(b) Training Data

Figure 3.1.1: Model output vs Target output for dataset2, $\lambda = 0$, $\sigma = 20$, $K = 24$

3.2 Dataset 3 : 2-music.txt

We applied feature scaling on the inputs to improve the model efficiency.

3.2.1 Without regularization

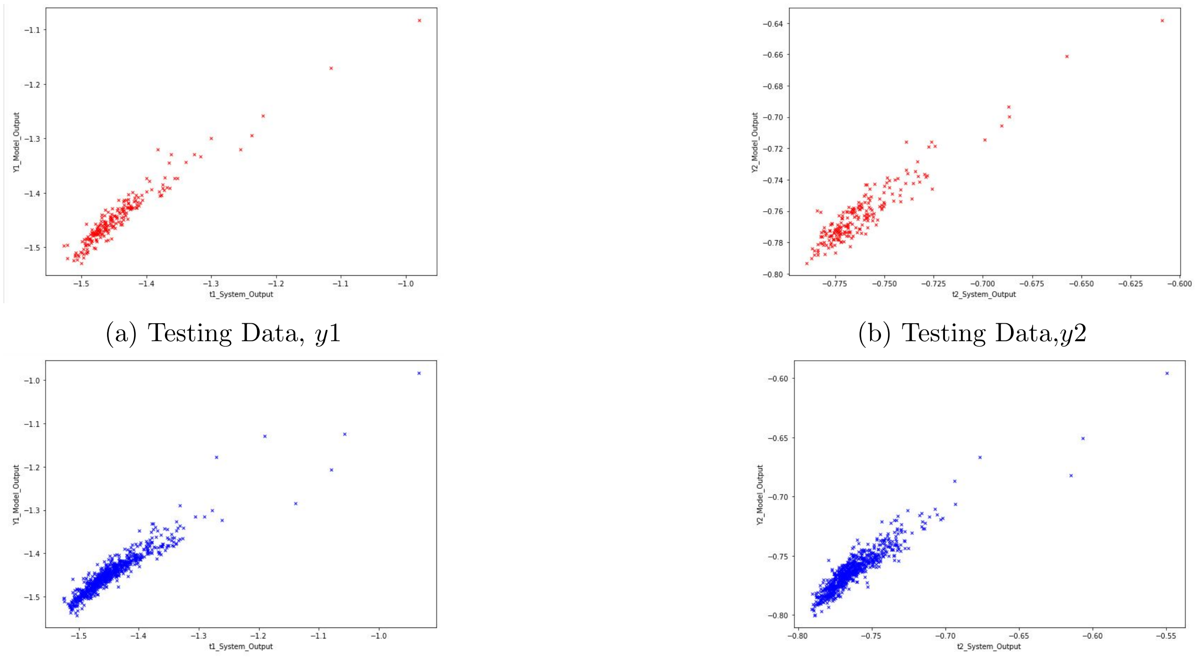
Hyperparameters : $\sigma = 250$; K = Number of clusters = $\{3, 5, 8, 10\}$

K	Training Dataset	Test Dataset	Validation Dataset
3	0.054272	0.062679	0.043885
5	0.047456	0.050827	0.036754
8	0.021510	0.023942	0.019601
10($Best$)	0.016433	0.015821	0.013432

Table 3.2.1: ERMS for dataset 3, $\lambda = 0$

Observation :

Here since we were using 70% of the data we had the liberty to go for a complex model i.e. for higher value of K with the performance of model still increasing at a good rate.



(a) Testing Data, $y1$

(b) Testing Data,$y2$

(c) Training Data,$y1$

(d) Training Data,$y2$

Figure 3.2.1: Model output vs Target output for dataset 3, $\lambda = 0$