Group 28 Report for Assignment 1

Pravin Ravi CE22B092
Abhiram Sulige AE23B059

Contents

1 Dataset 1 **3**
1.1 Linear Model for Regression using Polynomial Basis Functions 3
1.1.1 Erms Without Regularization on Train, Test and validation data . 4
1.1.2 Erms With Regularization on Train, Test and validation data and Degree = [7, 9], Sample size : 10 5
1.1.3 Conclusion: 6
1.1.4 Erms Without Regularization($\lambda = 0$) on Train, Test and validation data 6
1.1.5 Erms With Regularization on Train, Test and validation data and Degree = [7,9] 7
1.1.6 Conclusion: 8
1.2 Linear Model for Regression using Gaussian Basis Functions 9
1.2.1 Erms on Train, Test and validation data (Size - 10) 10
1.2.2 Erms on Train, Test and validation data (Size - 50) 10
1.3 Best Model 11

2 Dataset 2 **12**
2.1 Linear model for regression using polynomial basis functions 12
2.1.1 Erms Without Regularization on Train, Test and validation data . 12
2.1.2 Erms With Regularization on Train, Test and validation data and Degree = [6, 8] 13
2.1.3 Conclusion: 14
2.1.4 Erms Without Regularization($\lambda = 0$) on Train, Test and validation data 15
2.1.5 Erms With Regularization on Train, Test and validation data and Degree = [6, 8] 16
2.1.6 Conclusion: 17
2.2 Linear model for regression using Gaussian basis functions 18
2.2.1 Erms on Train, Test and validation data (Size - 25) 18
2.2.2 Erms on Train, Test and validation data (Size - 100) 18
2.3 Best Model 19

3 Dataset 3 **20**
3.1 Linear model for regression using polynomial basis functions 20
3.1.1 Erms Without Regularization on Train, Test and validation data . 20
3.2 Conclusion: 20
3.3 Linear model for regression using Gaussian basis functions 21
3.3.1 Erms on Train, Test and validation data 21
3.4 Best Model 21

1 Dataset 1

1.1 Linear Model for Regression using Polynomial Basis Functions

Polynomial basis function regression is an extension of linear regression, where the original input variable is mapped to a higher-dimensional space using polynomial transformations. This allows the model to capture non-linear relationships between the input and the target variable while still maintaining a linear structure in terms of the model parameters.

Univariate Polynomial Regression

In the case of univariate polynomial regression, suppose we have a single input variable $x \in R$ with n data points. For a polynomial of degree d, the feature vector corresponding to a single input x_i is expanded into the following basis:

$$\phi(x_i) = [1, x_i, x_i^2, \dots, x_i^d]^T \tag{1}$$

Stacking these vectors for all n data points gives the design matrix $\phi_{n\times(d+1)}$. The regression model can then be expressed as:

$$\mathbf{y}_{n\times 1} = \phi_{n\times(d+1)}\mathbf{W}_{(d+1)\times 1} \tag{2}$$

where $\mathbf{W}$ represents the weights corresponding to each polynomial term, including the bias.

Closed-form Solution

The optimal weight vector $\mathbf{W}$ is estimated by minimizing the mean squared error between the predicted and actual outputs. The closed-form solution, incorporating an optional regularization term λI, is given by:

$$\mathbf{W} = (\phi^T\phi + \lambda I)^{-1}\phi^T\mathbf{y} \tag{3}$$

Here:

- $\phi^T\phi$ represents the correlation between the polynomial features,
- λI is the regularization term that helps prevent overfitting (Ridge Regression),
- $\phi^T\mathbf{y}$ captures the correlation between the features and the target variable.

Multivariate Polynomial Regression

For the multivariate case, with m input variables $\mathbf{x} = [x_1, x_2, \dots, x_m]$, the polynomial basis functions are constructed by including all monomials up to degree d. For example, for $m = 2$ and $d = 2$, the feature mapping becomes:

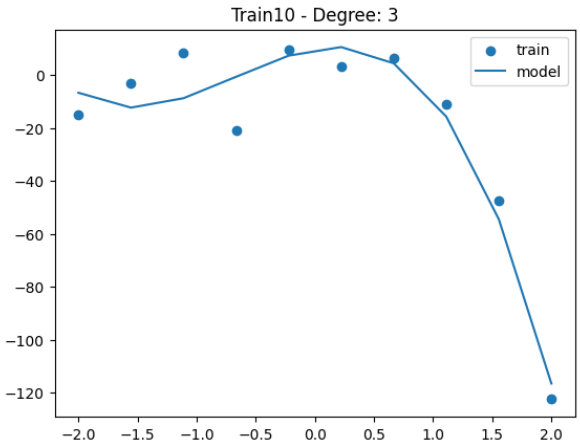
$$\phi(x_1, x_2) = [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]^T \tag{4}$$

In general, the number of features grows combinatorially with both m and d. The regression model and closed-form solution remain structurally the same as in the univariate case, but with a larger design matrix ϕ.

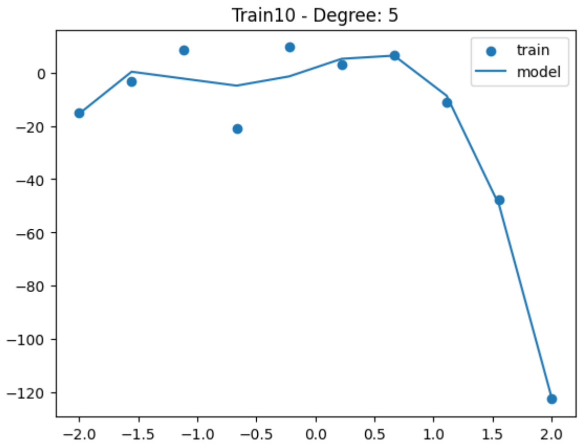
Training Dataset 1a (Size-10)

1.1.1 Erms Without Regularization on Train, Test and validation data

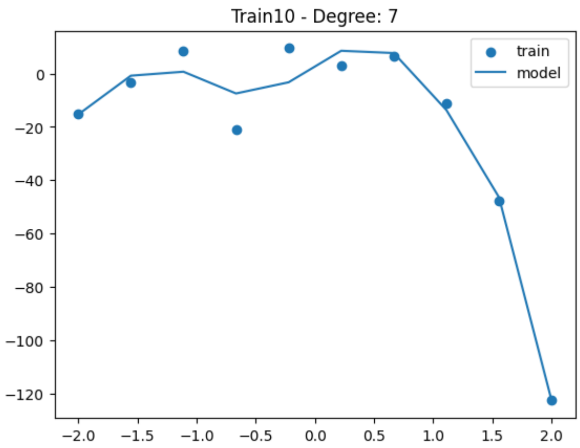
Degree	Train Erms	Validation Erms	Test Erms
3	10.12	13.73	13.37
5	7.20	13.60	11.22
7	6.69	13.80	11.40
9	0	39.71	39.53



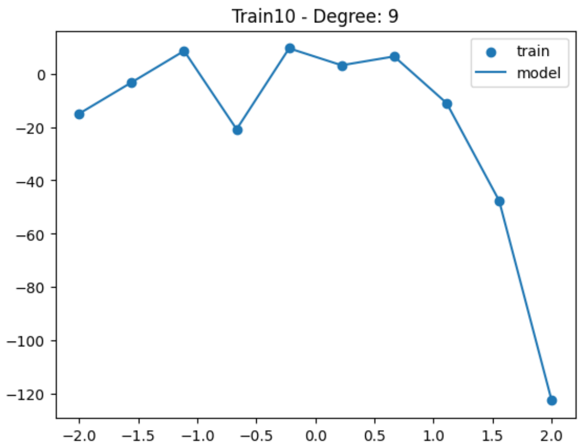
(a)



(b)



(c)



(d)

Figure 1: Curve fitting for different degree without regularization, Sample size: 10

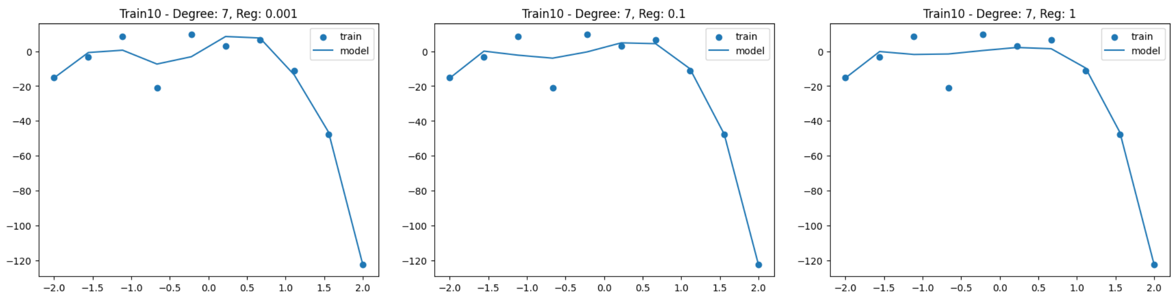
1.1.2 Erms With Regularization on Train, Test and validation data and Degree = [7, 9], Sample size : 10

Table 1: Erms With Regularization (Degree = 7)

λ	Train Erms	Validation Erms	Test Erms
0.001	6.69	13.78	11.36
0.1	7.20	13.40	10.84
1	7.75	13.28	10.70

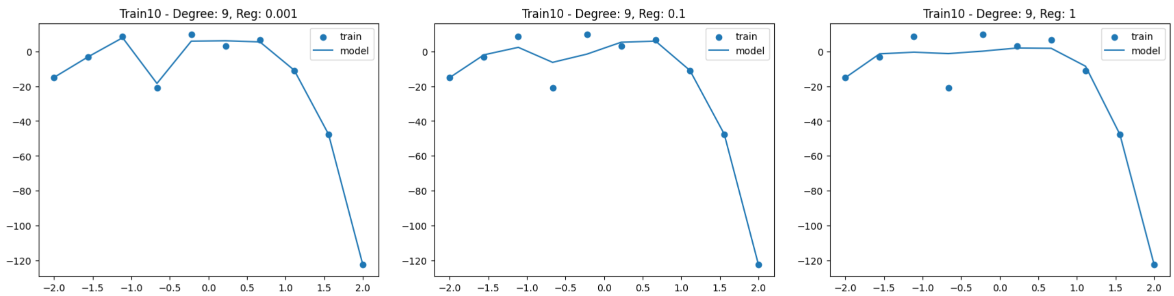
Table 2: Erms With Regularization (Degree = 9)

λ	Train Erms	Validation Erms	Test Erms
0.001	1.71	29.67	29.32
0.1	6.15	13.69	11.81
1	7.65	13.11	10.89



(a) (b) (c)

Figure 2: Curve fitting with regularization for degree=7, Sample size: 10.



(a) (b) (c)

Figure 3: Curve fitting with regularization for degree=9, Sample size: 10.

We see that the best fit for the data is obtained for degree = 9 and $\lambda = 1$ model, it gives the best performance (least ERMS in test data).

1.1.3 Conclusion:

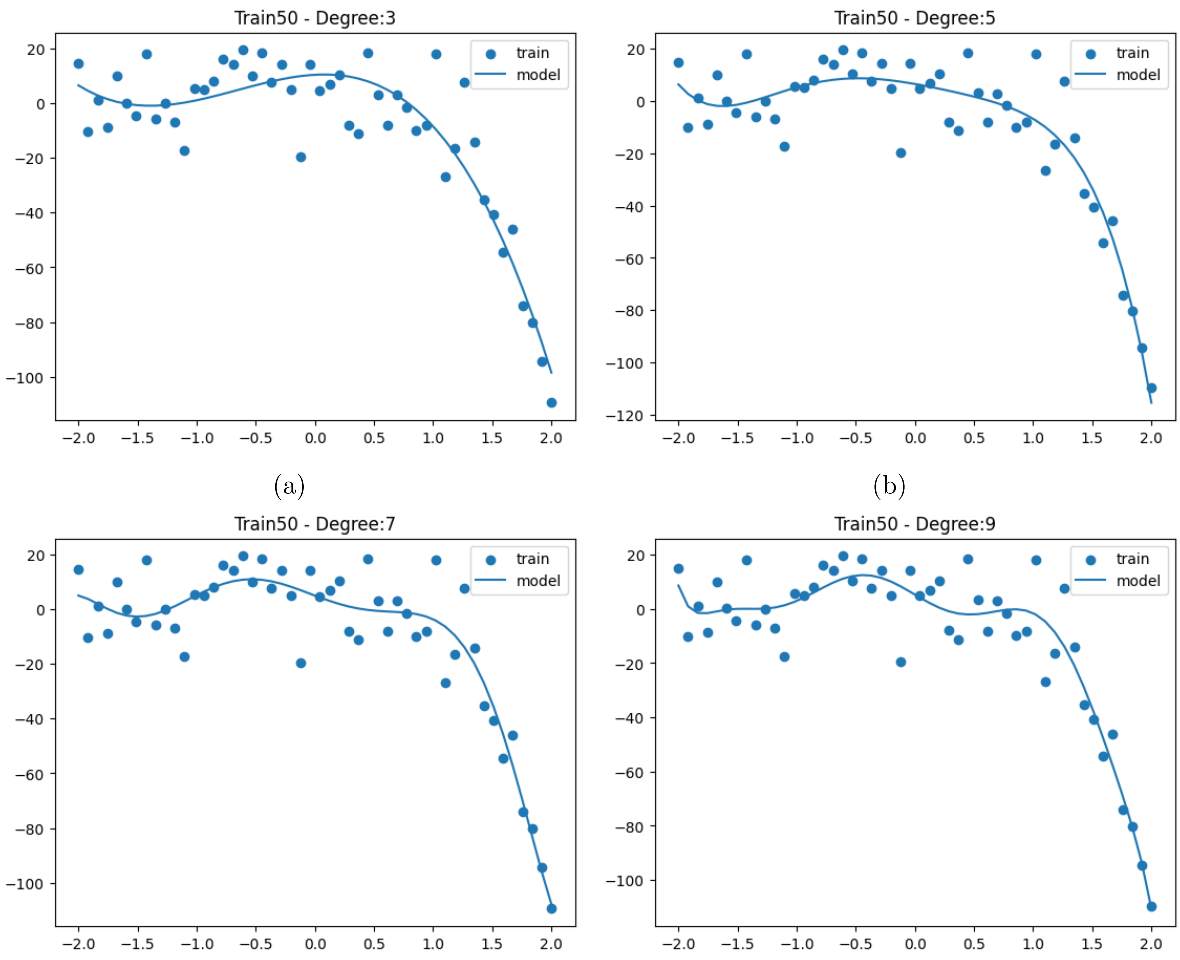
ERMS for best performing model: (M = 9 and $\lambda = 1$)
Train Data Erms: 7.65
Test Data Erms: 10.89

Training Dataset 1b (Size-50)

1.1.4 Erms Without Regularization($\lambda = 0$) on Train, Test and validation data

Table 3: Erms Without Regularization (Sample size: 50)

Degree	Train Erms	Validation Erms	Test Erms
3	11.42	14.40	12.61
5	10.34	13.91	11.59
7	10.06	13.74	12.14
9	9.89	14.02	12.03



(a) (b) (c) (d)

Figure 4: Curve fitting for different degree without regularization, Sample size: 50

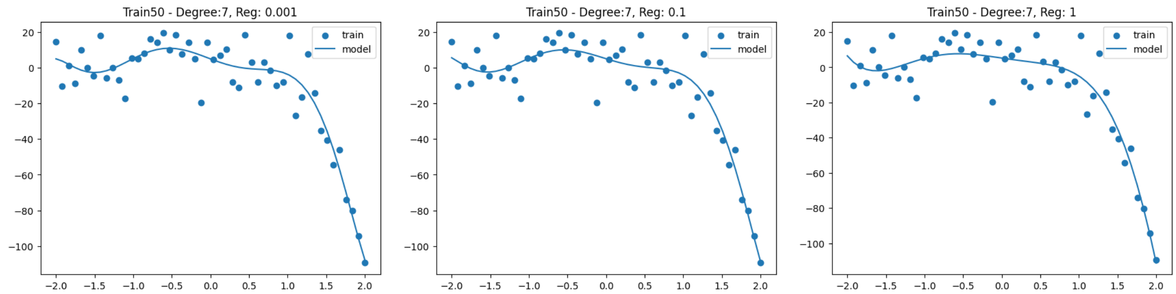
1.1.5 Erms With Regularization on Train, Test and validation data and Degree = [7,9]

Table 4: Erms With Regularization (Degree = 7, Sample size: 50)

λ	Train Erms	Validation Erms	Test Erms
0.001	10.05	13.74	12.13
0.1	10.07	13.66	11.96
1	10.24	13.55	11.58

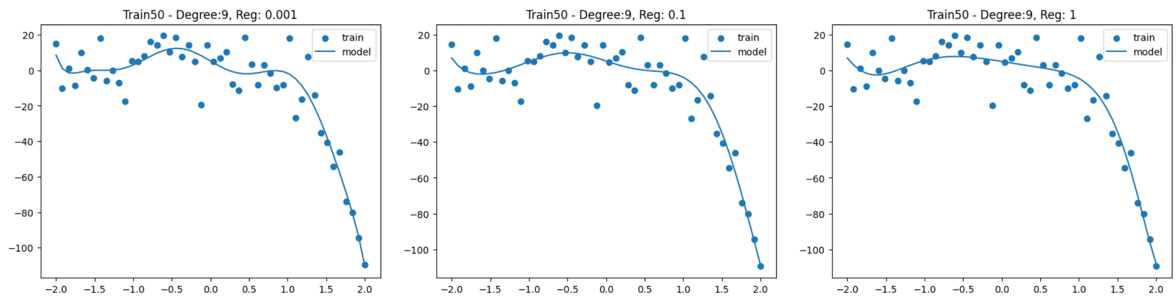
Table 5: Erms With Regularization (Degree = 9, Sample size: 50)

λ	Train Erms	Validation Erms	Test Erms
0.001	9.90	14.01	12.02
0.1	10.03	13.66	11.92
1	10.22	13.45	11.81



(a) (b) (c)

Figure 5: Curve fitting with regularization for degree=7, Sample size: 50.



(a) (b) (c)

Figure 6: Curve fitting with regularization for degree=9, Sample size: 50.

****We see that the best fit for the data is obtained for degree: 6 and when we analyze (using validation ERMS) the model for given λ values and different Degree (given M values) we see that for degree = 6 and $\lambda = 1$ model gives the best performance (least ERMS in test data).*******

1.1.6 Conclusion:

ERMS for best performing model: (M = 7 and $\lambda = 1$)
Train Data Erms: 10.24
Test Data Erms: 11.58

1.2 Linear Model for Regression using Gaussian Basis Functions

Gaussian basis function regression is another approach to extend linear regression, where the input variable is mapped to a new feature space using localized Gaussian functions. Unlike polynomial basis functions, which grow unbounded with increasing input, Gaussian basis functions create localized "bumps" centered around predefined means, allowing the model to capture local variations in the data.

Univariate Gaussian Basis Regression

In the case of univariate regression, suppose we have an input variable $x \in R$ with n data points. For M Gaussian basis functions, each basis is defined as:

$$\phi_j(x) = \exp\left(-\frac{||x - \mu_j||^2}{\sigma}\right), \quad j = 1, 2, \ldots, M \tag{5}$$

where μ_j is the center of the j^{th} Gaussian and σ controls the spread (width).

Stacking these for all n data points gives the design matrix $\phi_{n \times M}$. The regression model can then be expressed as:

$$\mathbf{y}_{n\times 1} = \phi_{n\times M}\mathbf{W}_{M\times 1} \tag{6}$$

where $\mathbf{W}$ represents the weights associated with each Gaussian function.

Closed-form Solution

As in the polynomial case, the optimal weight vector $\mathbf{W}$ is obtained by minimizing the mean squared error. The closed-form solution, with an optional regularization term λI, is given by:

$$\mathbf{W} = (\phi^T\phi + \lambda I)^{-1}\phi^T\mathbf{y} \tag{7}$$

Here:

- $\phi^T\phi$ represents the overlap between Gaussian basis functions,
- λI provides regularization to control overfitting,
- $\phi^T\mathbf{y}$ captures the alignment between the basis functions and the target values.

Multivariate Gaussian Basis Regression

For the multivariate case, with m input variables $\mathbf{x} = [x_1, x_2, \ldots, x_m]$, each Gaussian basis function is defined as:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{||\mathbf{x} - \boldsymbol{\mu}_j||^2}{\sigma}\right) \tag{8}$$

where $\boldsymbol{\mu}_j \in R^m$ is the center of the j^{th} Gaussian in the m-dimensional space. The design matrix $\phi_{n\times M}$ is constructed by evaluating all M Gaussians at each of the n input points. The regression model and closed-form weight solution remain structurally identical to the univariate case.

Training Dataset 1a (Size - 10)

1.2.1 Erms on Train, Test and validation data (Size - 10)

No. of Basis Functions	Train Erms	Validation Erms	Test Erms
1	37.76	29.01	31.04

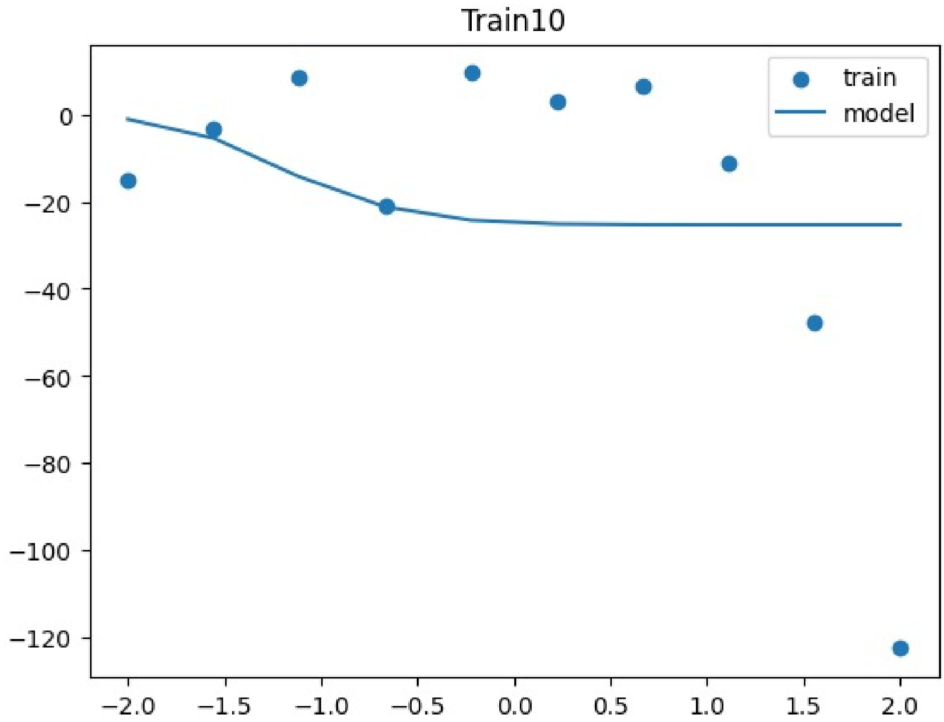
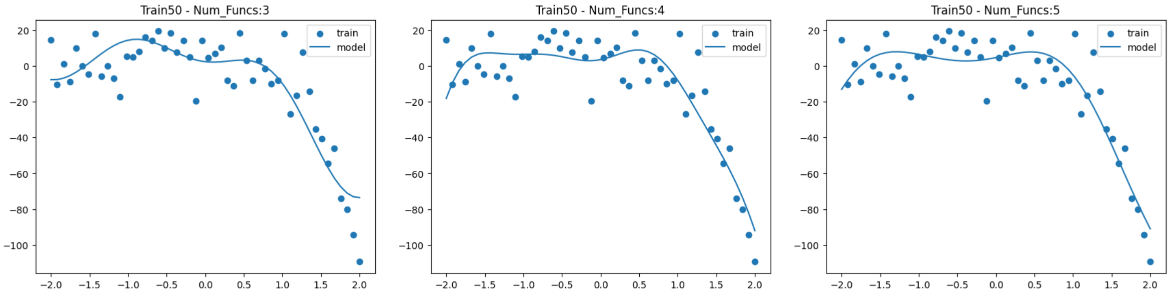


Figure 7: Curve fitting for number of basis functions = 1, Sample size: 10

Training Dataset 1b (Size - 50)

1.2.2 Erms on Train, Test and validation data (Size - 50)

No. of Basis Functions	Train Erms	Validation Erms	Test Erms
3	10.22	16.03	15.11
4	10.22	15.22	12.41
5	10.22	14.54	12.13



(a) (b) (c)

Figure 8: Curve fitting for various number of basis functions, Sample size: 50.

1.3 Best Model

Polynomial degree 9 with lambda = 1

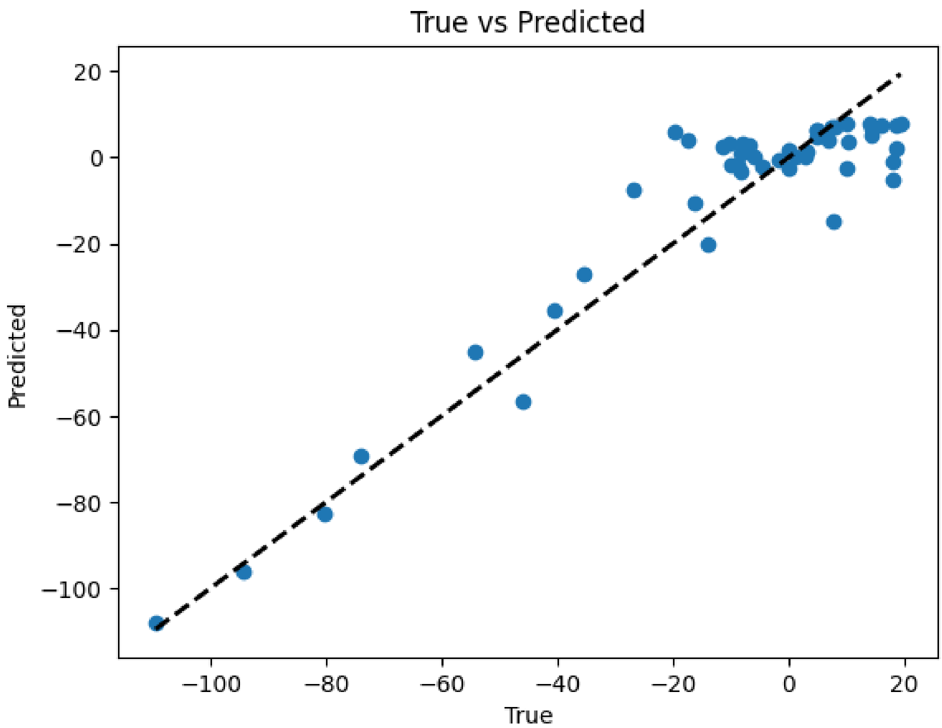


Figure 9: Best fit visualization for dataset1

2 Dataset 2

2.1 Linear model for regression using polynomial basis functions

Training Dataset 2a (Size-25)

2.1.1 Erms Without Regularization on Train, Test and validation data

Degree	Train Erms	Validation Erms	Test Erms
2	12.83	38.76	39.24
4	1.05	28.73	40.91
6	153.91	257.12	314.37
8	281.63	5835.30	6935.00

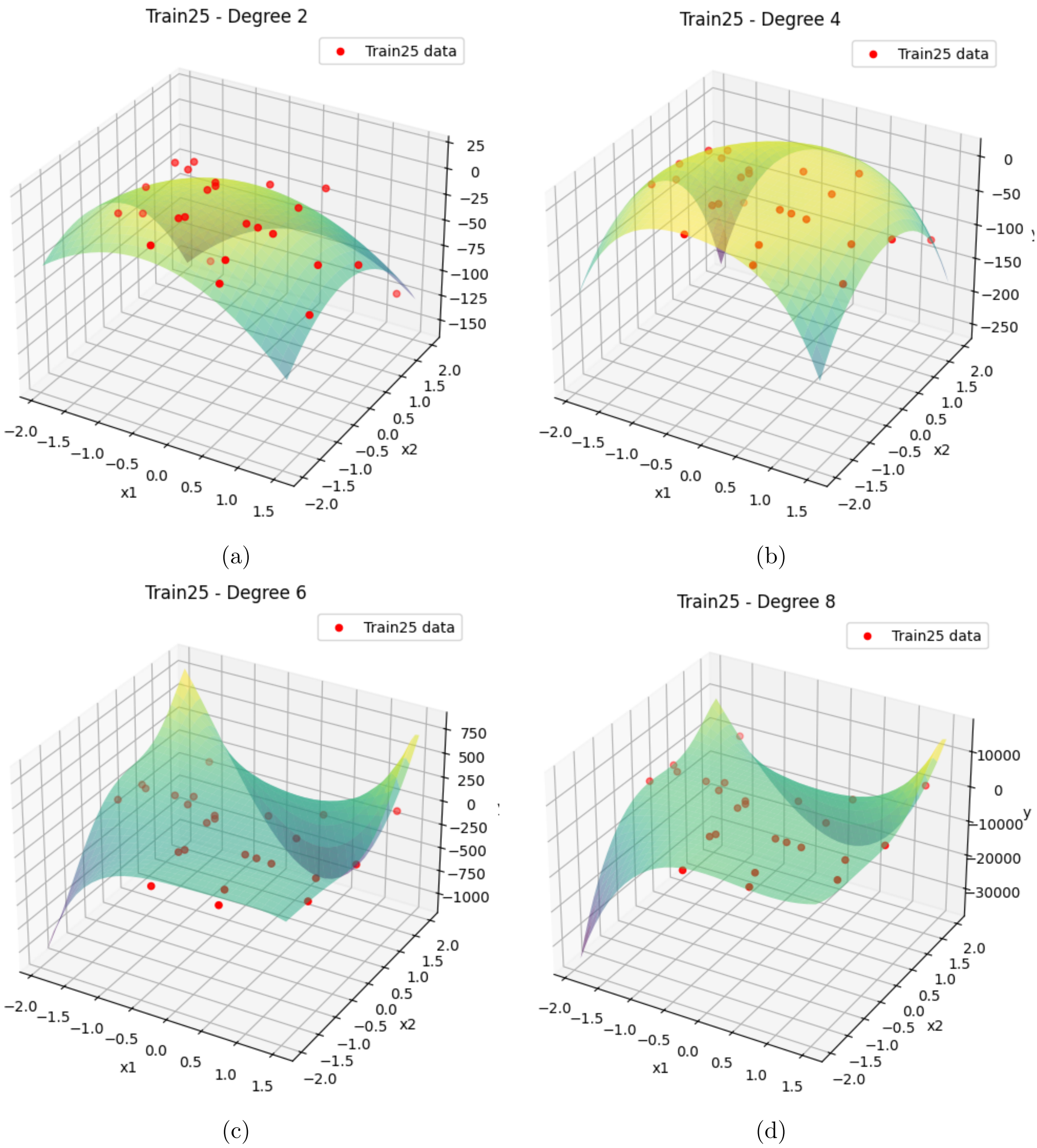


Figure 10: Surface plot for different degree without regularization, Sample size: 25

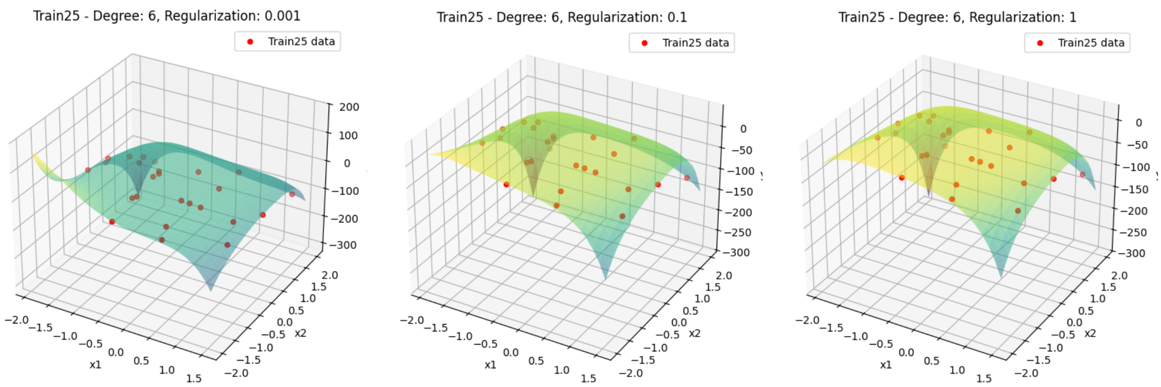
2.1.2 Erms With Regularization on Train, Test and validation data and Degree = [6, 8]

Table 6: Erms With Regularization (Degree = 6)

λ	Train Erms	Validation Erms	Test Erms
0.001	0.02	4.95	4.68
0.1	0.08	9.34	16.67
1	0.26	11.84	21.29

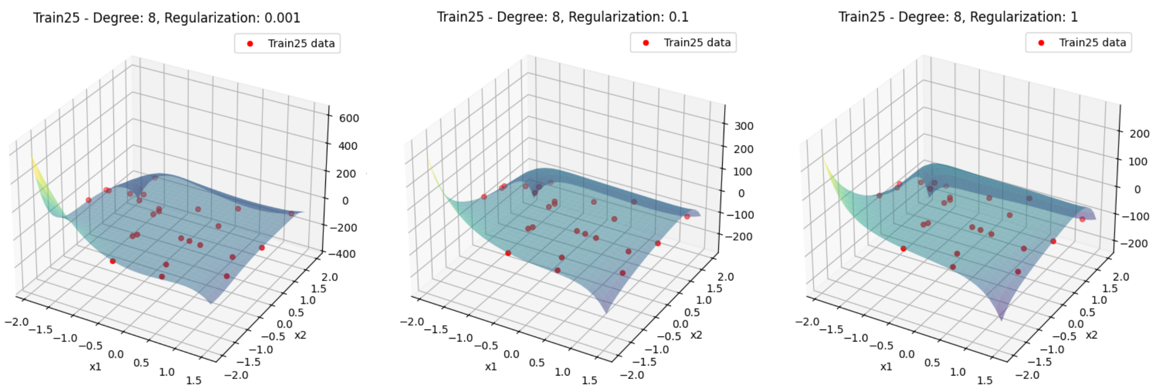
Table 7: Erms With Regularization (Degree = 8)

λ	Train Erms	Validation Erms	Test Erms
0.001	0.01	21.04	43.26
0.1	0.11	18.66	21.27
1	0.31	22.50	19.10



(a) (b) (c)

Figure 11: Curve fitting with regularization for degree=6, Sample size: 25.



(a) (b) (c)

Figure 12: Curve fitting with regularization for degree=8, Sample size: 25.

We see that the best fit for the data is obtained for degree: 4 and when we analysis (using validation data ERMS) the model for given λ values and different Degree (given M values). we see that for degree = 4 and $\lambda = 1$ model gives the best performance (least ERMS in test data)

2.1.3 Conclusion:

ERMS for best performing model: (M = 6 and $\lambda = 0.001$)
Train Data Erms: 0.02
Test Data Erms: 4.68

Training Dataset 2b (Size-100)

2.1.4 Erms Without Regularization($\lambda = 0$) on Train, Test and validation data

Table 8: Erms Without Regularization (Sample size: 100)

Degree	Train Erms	Validation Erms	Test Erms
2	24.14	32.22	27.52
4	4.48	10.23	7.01
6	0.08	0.14	0.13
8	0.07	0.37	0.29

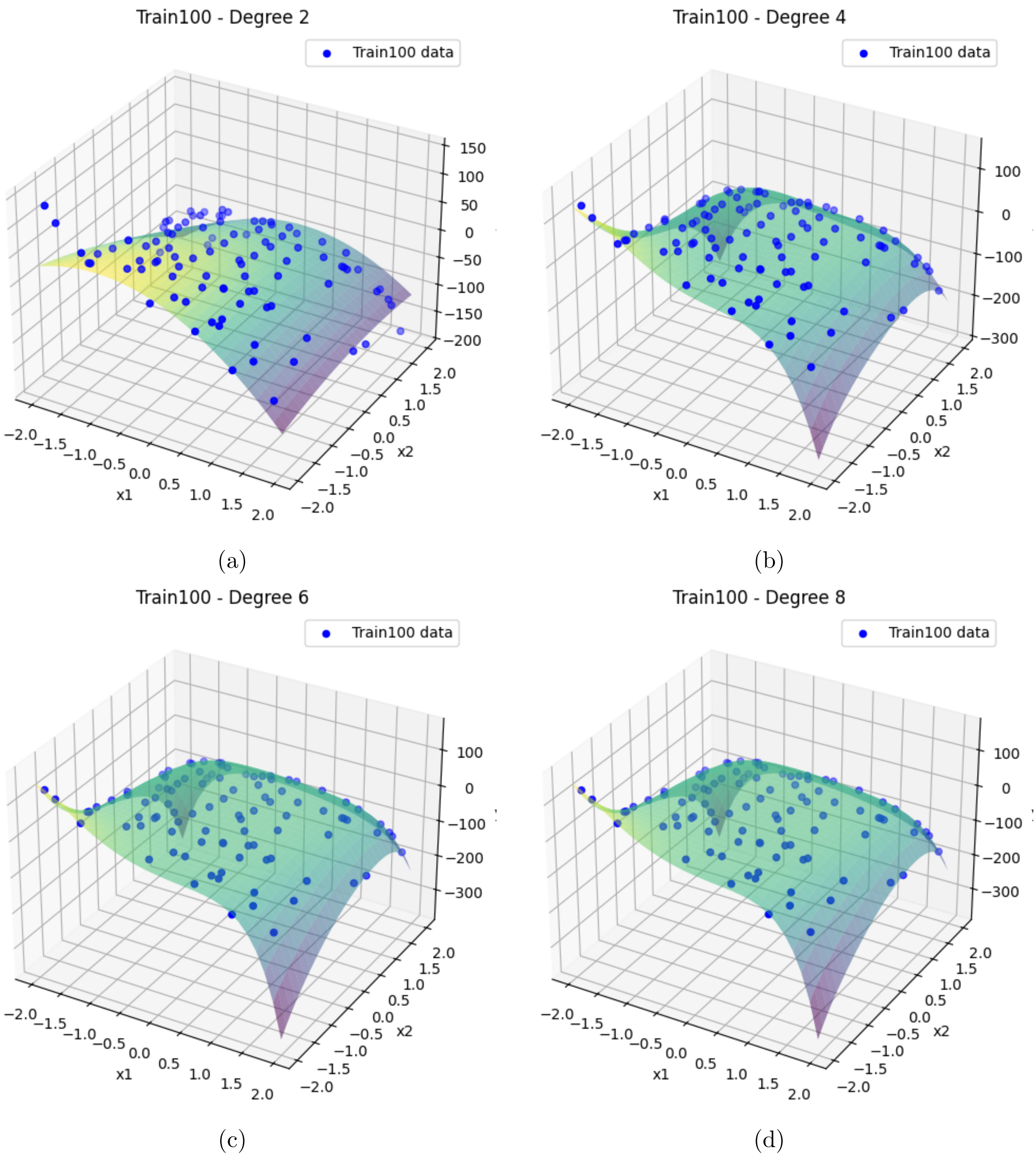


Figure 13: Surface plot for different degree without regularization, Sample size: 100

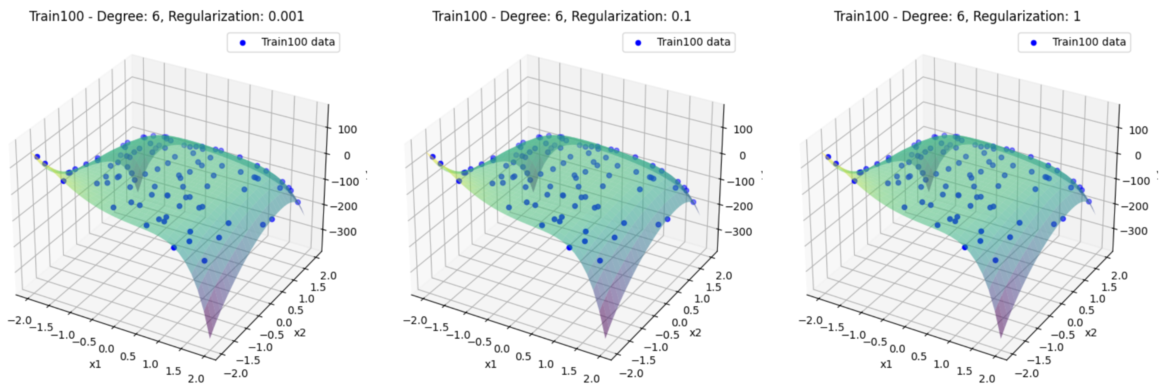
2.1.5 Erms With Regularization on Train, Test and validation data and Degree = [6, 8]

Table 9: Erms With Regularization (Degree = 6, Sample size: 100)

λ	Train Erms	Validation Erms	Test Erms
0.001	0.08	0.14	0.13
0.1	0.09	0.13	0.14
1	0.19	0.33	0.28

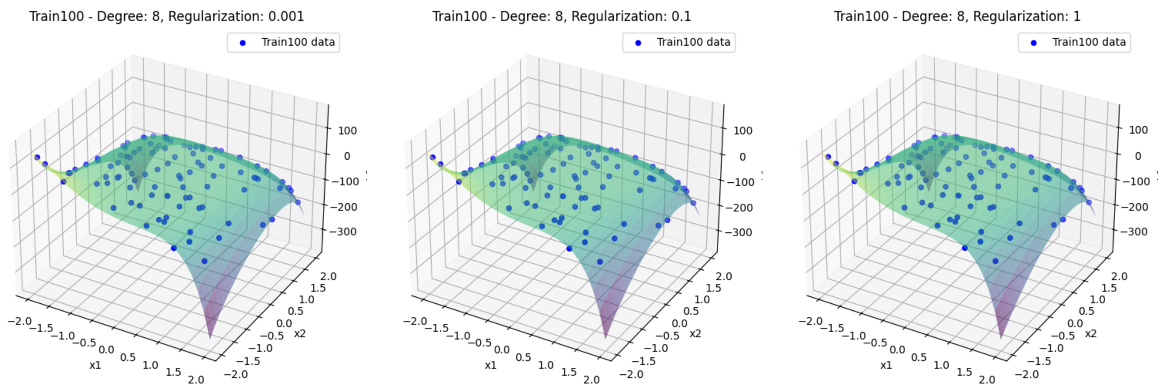
Table 10: Erms With Regularization (Degree = 8, Sample size: 100)

λ	Train Erms	Validation Erms	Test Erms
0.001	0.07	0.37	0.29
0.1	0.09	0.69	0.41
1	0.20	1.65	0.92



(a) (b) (c)

Figure 14: Surface plots with regularization for degree=6, Sample size: 100.



(a) (b) (c)

Figure 15: Surface plots with regularization for degree=8, Sample size: 100.

We see that the best fit for the data is obtained for degree: 6 and when we analyze (using validation data ERMS) the model for given λ values and different degrees (given M values). we see that for degree = 6 and for $\lambda = 0.1$ values, model gives the best performance (least ERMS for test data)

2.1.6 Conclusion:

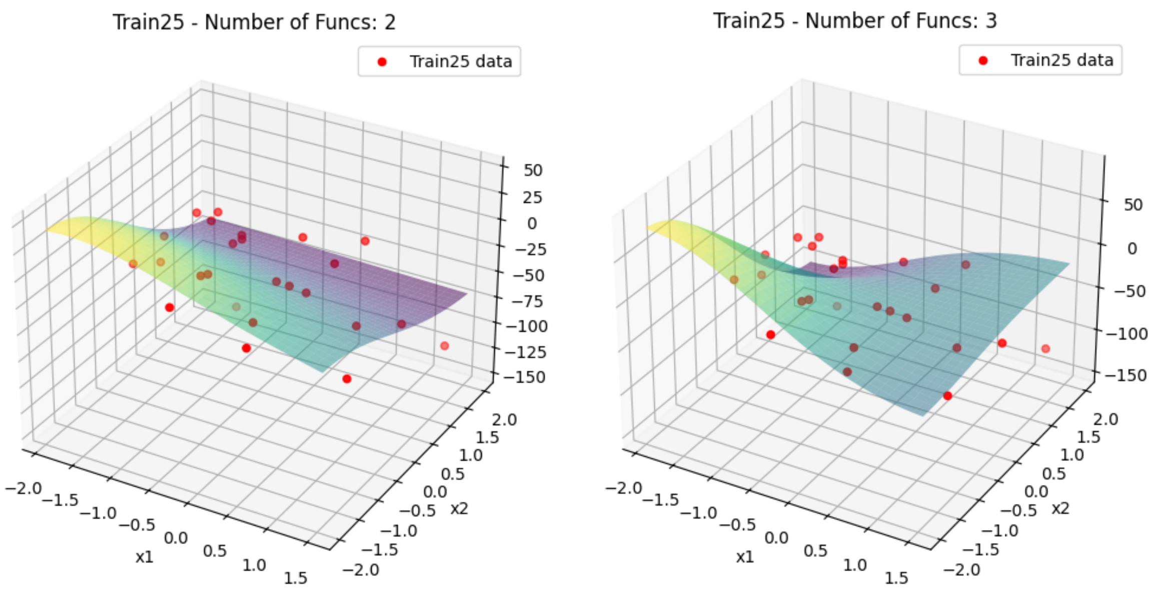
ERMS for best performing model: (M = 6 and $\lambda = 0.1$)
Train Data Erms: 0.09
Test Data Erms: 0.14

2.2 Linear model for regression using Gaussian basis functions

Training Dataset 2a (Size - 25)

2.2.1 Erms on Train, Test and validation data (Size - 25)

No. of Basis Functions	Train Erms	Validation Erms	Test Erms
2	31.28	51.86	43.34
3	30.77	51.78	44.37



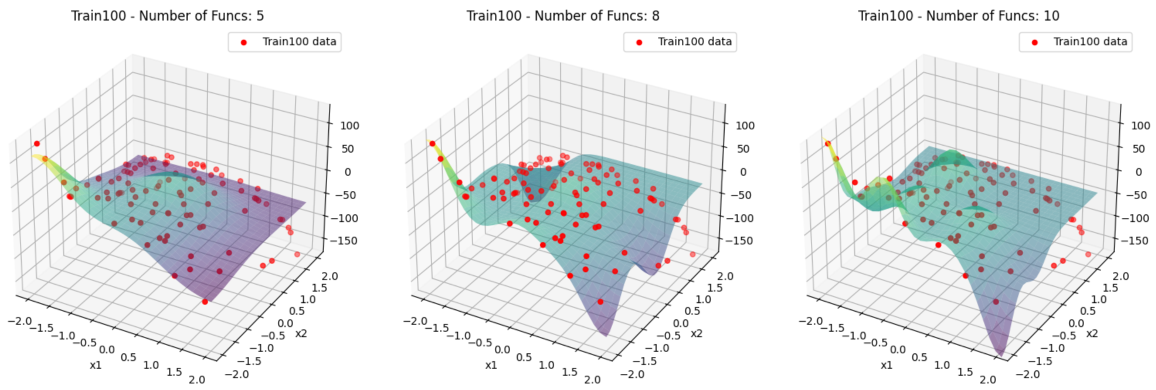
(a) (b)

Figure 16: Curve fitting for various number of basis functions, Sample size: 25.

Training Dataset 2b (Size - 100)

2.2.2 Erms on Train, Test and validation data (Size - 100)

No. of Basis Functions	Train Erms	Validation Erms	Test Erms
5	27.80	38.50	30.33
8	29.48	39.52	31.82
10	30.88	36.64	27.19



(a) (b) (c)

Figure 17: Surface plot for different no. of basis functions, Sample size: 100

2.3 Best Model

Polynomial degree 6 with lambda = 0.1

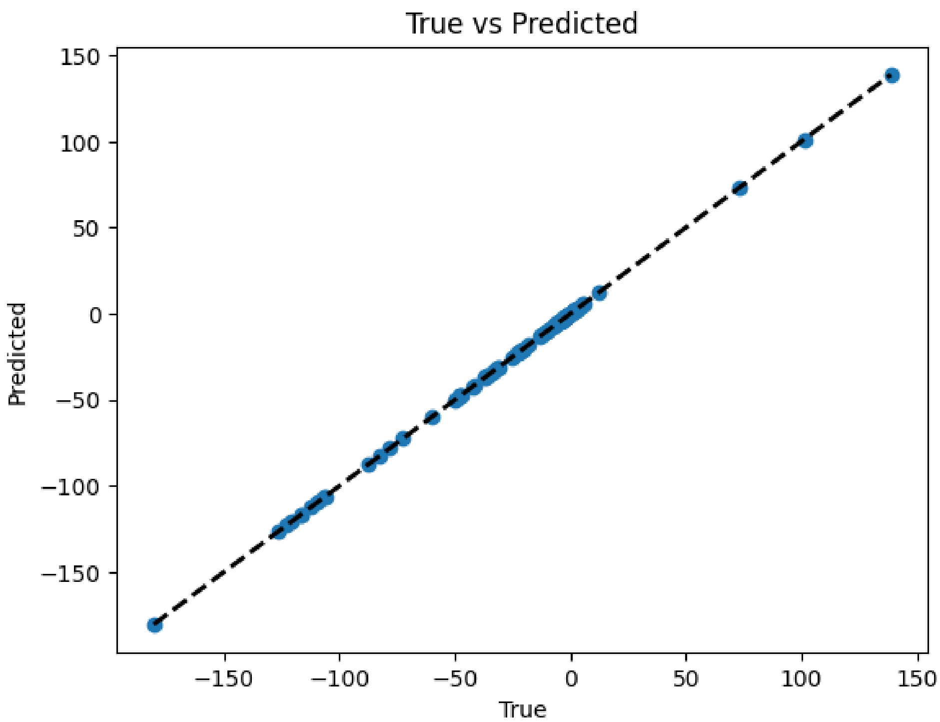


Figure 18: Best fit visualization for dataset2

3 Dataset 3

3.1 Linear model for regression using polynomial basis functions

Training Dataset 3 (Size-350)

3.1.1 Erms Without Regularization on Train, Test and validation data

Degree	Train Erms	Validation Erms	Test Erms
2	0.3805	0.3687	0.4087
3	0.3729	0.3700	0.4121
4	0.3662	0.3807	0.4125

3.2 Conclusion:

No Regularization was needed in this dataset as all 3 models did not demonstrate overfitting.

ERMS for best performing model: (M = 2)
Train Data Erms: 0.3805
Test Data Erms: 0.4087

3.3 Linear model for regression using Gaussian basis functions

Training Dataset 3 (Size - 350)

3.3.1 Erms on Train, Test and validation data

No. of Basis Functions	Train Erms	Validation Erms	Test Erms
20	0.46	0.48	0.43
25	0.46	0.49	0.43
30	0.46	0.49	0.43
35	0.45	0.49	0.43

3.4 Best Model

1) Polynomial degree 2 with no regression
2) Gaussian with num of funcs = 20

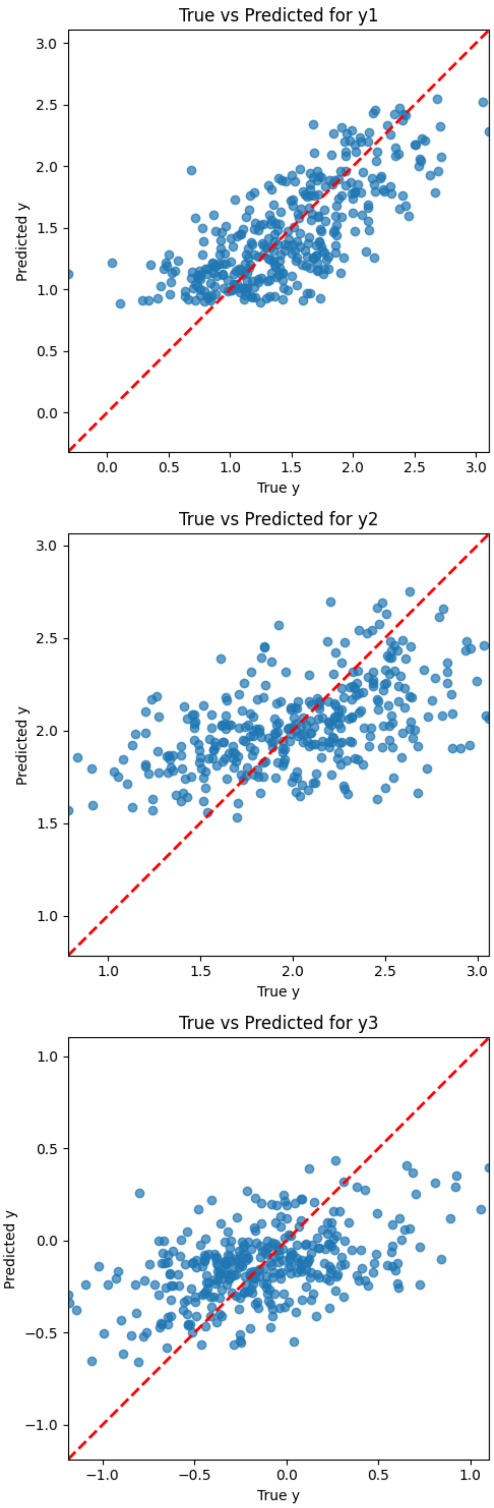


Figure 19: Best fit visualization for dataset3