Pattern Recognition and Machine Learning

CS5691

Assignment 1 Report

Team 16

Members:

CS23B097 Deepanjan Das

CS23B101 Ajoy Mathew

September 3, 2025

Contents

Abstract

This report presents our work for Assignment 1 of CS5691 (Pattern Recognition and Machine Learning). We analyze the given datasets, apply polynomial regression with and without regularization, and evaluate models based on training, validation, and test errors. The report compares models of different degrees and regularization strengths, visualizes curve fitting results, and identifies the best-performing models based on generalization performance.

1 Polynomial Regression (Multivariate)

For m-dimensional inputs $x \in \mathbb{R}^m$ we define a polynomial feature map

$$\varphi : \mathbb{R}^m \to \mathbb{R}^p,$$

which maps each raw input $x = (x_1, \ldots, x_m)$ to a vector of polynomial basis functions (terms) up to the chosen degree d. The exact number of basis functions p depends on whether you include only pure powers or also cross-terms (interaction terms). For the full polynomial basis (including all monomials of total degree $\leq d$) the number of features is

$$p \;=\; \binom{m+d}{d},$$

which includes the constant term 1.

Feature vector (single sample)

For a single input x, the feature (row) vector is written as

$$\varphi(x) \;=\; \begin{bmatrix}\phi_1(x) & \phi_2(x) & \cdots & \phi_p(x)\end{bmatrix}^\top \in \mathbb{R}^{p\times 1},$$

where each $\phi_j(x)$ is a polynomial basis function (for example x_1, $x_1.x_2$, x_3^2, etc.).

Design matrix

For n training samples $\{(x_i, y_i)\}_{i=1}^n$ build the design matrix

$$\Phi \;=\; \begin{bmatrix}\varphi(x_1)^\top \\ \varphi(x_2)^\top \\ \vdots \\ \varphi(x_n)^\top\end{bmatrix} \in \mathbb{R}^{n\times p},$$

so that each row is the feature row-vector for one sample.

Model (vector form)

The target vector $\mathbf{y} \in \mathbb{R}^n$ is modeled using a linear model in feature space:

$$\mathbf{y}_{n\times 1} \approx \Phi_{n\times p}\, \omega^*_{p\times 1},$$

where ω^* contains the model parameters (one weight per basis function).

Closed-form solution

The ordinary least-squares estimate (no regularization) is obtained by minimizing the squared error:

$$\omega^* \;=\; \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{y},$$

provided $\Phi^\top\Phi$ is invertible.

Prediction

For a new input x_{new} the feature row-vector $\varphi(x_{\text{new}}) \in \mathbb{R}^{1\times p}$ gives the predicted scalar output

$$\hat{y} \;=\; \varphi(x_{\text{new}})\, \omega^*.$$

Example: $m = 2, d = 2$ (full quadratic basis)

For two inputs $x = (x_1, x_2)$ and polynomial degree $d = 2$, the feature vector (including bias) can be written as

$$\varphi(x) = \begin{bmatrix}1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2\end{bmatrix},$$

so here $p = \binom{2+2}{2} = 6$. The design matrix Φ stacks such rows for all n samples, and the same closed-form $W^* = (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{y}$ applies.

2 Gaussian Regression

Gaussian regression generalizes polynomial regression by transforming the input data into Gaussian radial basis features. Instead of polynomial terms, the features are derived from Gaussian kernels centered at representative points in the input space.

K-means for Basis Selection

To determine the centers μ_j of the Gaussian basis functions, we use the K-means clustering algorithm. K-means partitions the input dataset into k clusters, where each cluster is represented by a centroid. These centroids are then used as the Gaussian centers μ_j.
The algorithm works as follows:

1. Initialize k cluster centers by selecting k random points from the dataset.
2. Assign each data point to the nearest cluster center (using Euclidean distance).
3. Update each cluster center as the mean of the points assigned to that cluster.
4. Repeat steps 2–3 until convergence (i.e., the cluster centers no longer change significantly).

Once the centers $\{\mu_1, \mu_2, \ldots, \mu_k\}$ are determined, the width parameter σ can be chosen heuristically (e.g., as the average distance between cluster centers).

Design Matrix

For n samples and k Gaussian basis functions, we construct the design matrix $\Phi \in \mathbb{R}^{n\times(k+1)}$ as

$$\Phi = \begin{bmatrix}1 & \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_k(x_1) \\ 1 & \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_k(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(x_n) & \phi_2(x_n) & \cdots & \phi_k(x_n)\end{bmatrix},$$

where the constant column of ones accounts for the bias term w_0, and each Gaussian basis function is defined as

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

Model

The target vector is modeled as

$$\vec{y}_{n\times 1} \approx \Phi_{n\times(k+1)}\omega^*_{(k+1)\times 1},$$

where

$$\omega^* = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_k \end{bmatrix}$$

is the vector of model parameters.

Closed-form Solution

The least squares solution for the optimal parameters is given by

$$\omega^* = \left(\Phi^T\Phi\right)^{-1}\Phi^T\vec{y}.$$

Prediction

For a new input x_{new}, the corresponding feature vector is

$$\phi(x_{\text{new}}) = \begin{bmatrix} 1 & \phi_1(x_{\text{new}}) & \phi_2(x_{\text{new}}) & \cdots & \phi_k(x_{\text{new}}) \end{bmatrix},$$

and the predicted output is

$$\hat{y} = \phi(x_{\text{new}})\omega^*.$$

3 Dataset1

3.1 Description

Input data is 1-dimensional (Univariate).

1. Training Dataset 1(a): 10 examples
2. Training Dataset 1(b): 50 examples

Training Dataset 1(a) and 1(b)

3.2 Using Polynomial Basis Functions

3.2.1 Results and Plots(Without Regularization)

Table 1 summarizes the ERMS values for the training and validation sets at different polynomial degrees.

Table 1: ERMS values on Train and Validation data (without regularization).

Degree	Train(1a)	Valid(1a)	Train(1b)	Valid(1b)
3	16.3023	16.6348	10.5502	15.4843
5	10.7738	12.8568	9.2017	13.7494
7	9.8478	13.4119	9.1073	13.6367
9	2.2e-08	41.7797	9.0117	13.7992

The following figures show polynomial curve fitting results for different degrees. Each plot compares the training data against the fitted polynomial curve.

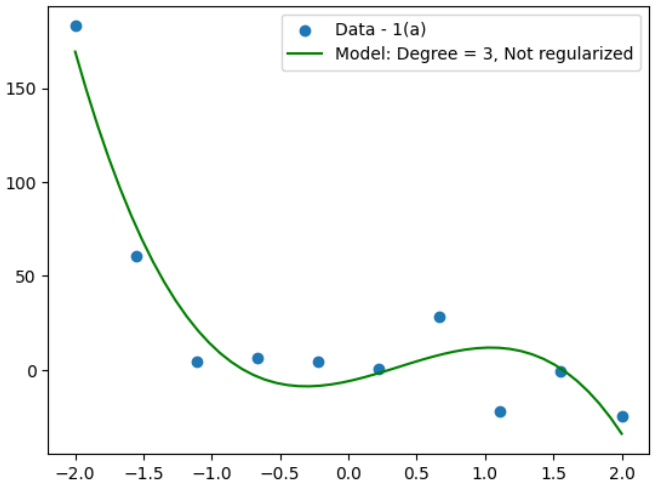

Figure 1: Polynomial Curve Fitting 1(a) (Degree = 3).

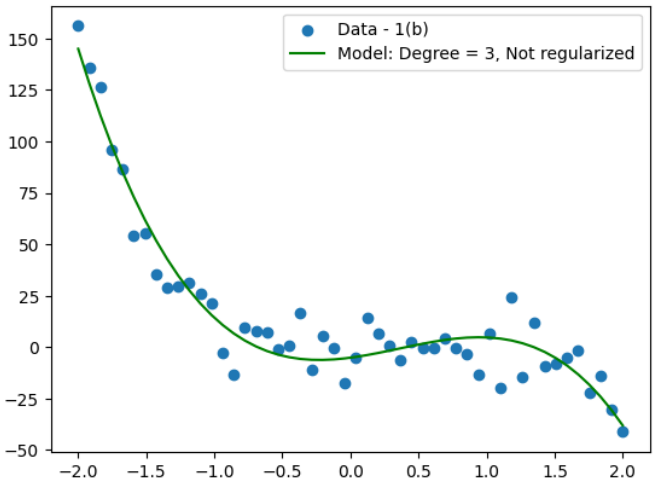

Figure 2: Polynomial Curve Fitting 1(b) (Degree = 3).

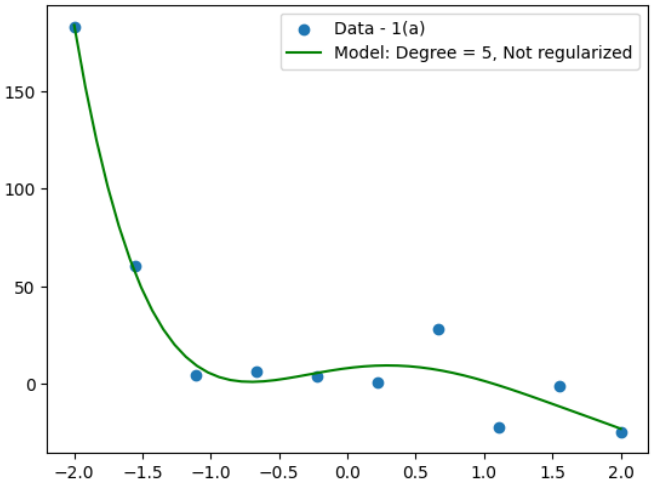

Figure 3: Polynomial Curve Fitting 1(a) (Degree = 5).

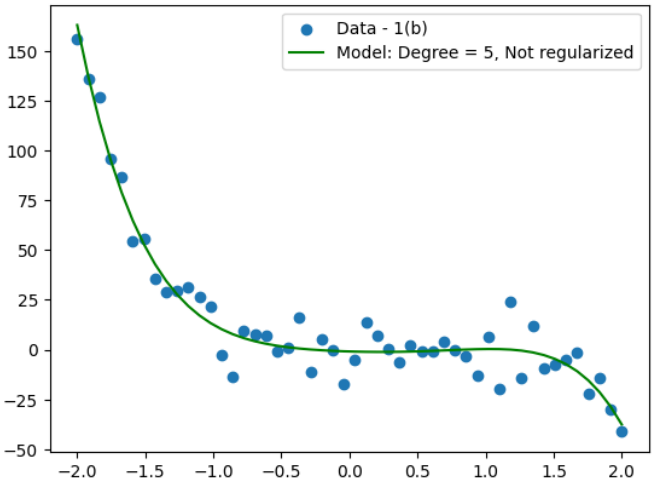

Figure 4: Polynomial Curve Fitting 1(b) (Degree = 5).

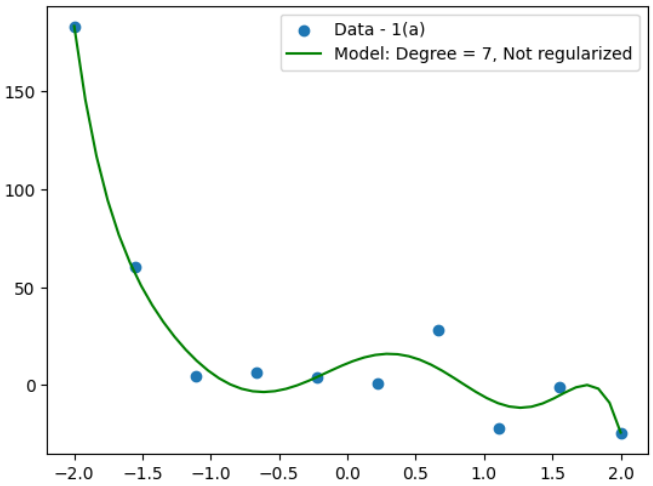

Figure 5: Polynomial Curve Fitting 1(a) (Degree = 7).

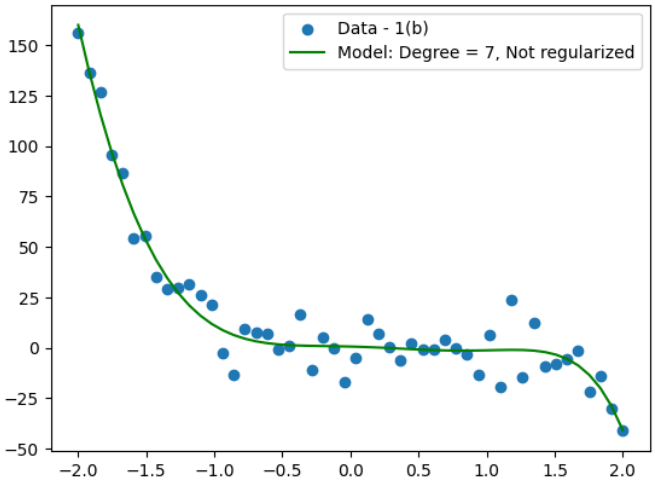

Figure 6: Polynomial Curve Fitting 1(b) (Degree = 7).

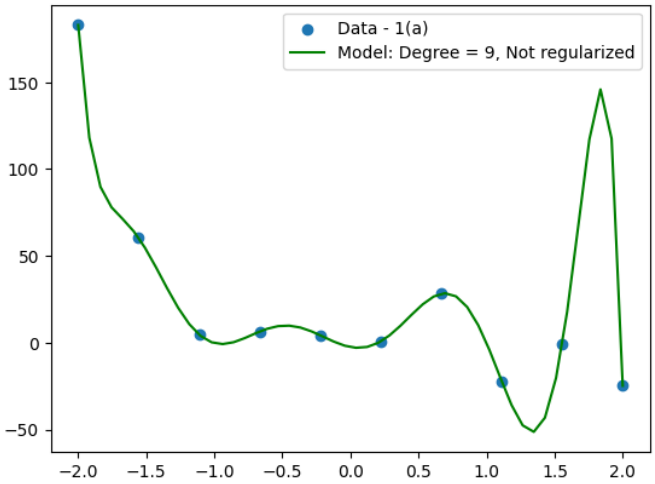

Figure 7: Polynomial Curve Fitting 1(a) (Degree = 9).

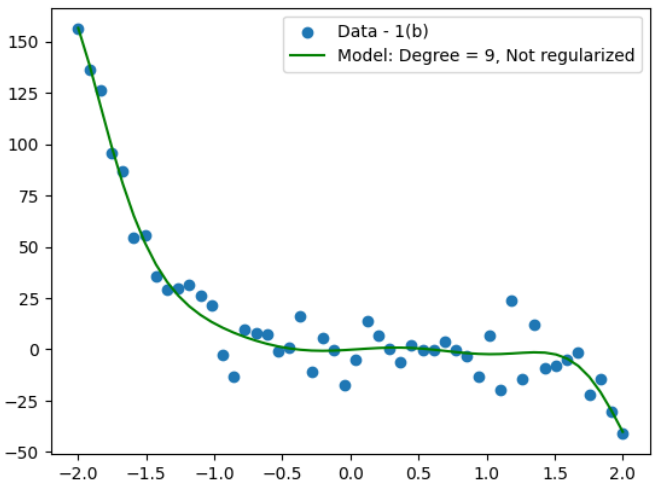

Figure 8: Polynomial Curve Fitting 1(b) (Degree = 9).

3.2.2 Applying Regularization

Dataset - 1(a): It turned out, there was no overfitting for models with degree $= 3, 5$ and 7, given an **overfitting ratio threshold of** 1.5, when compared with the **RMSE** from validation data. Following are the plots after applying regularization to models of degree 9, for different values of λ, from which we choose the best model.

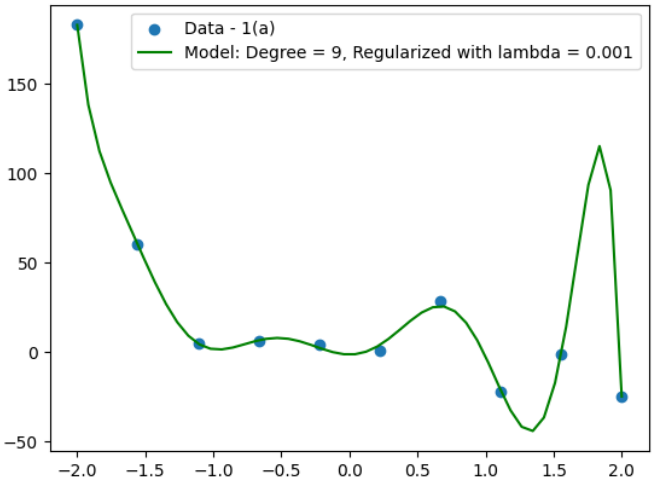

Figure 9: Regularizing with $\lambda = 0.001$ (Degree $= 9$).

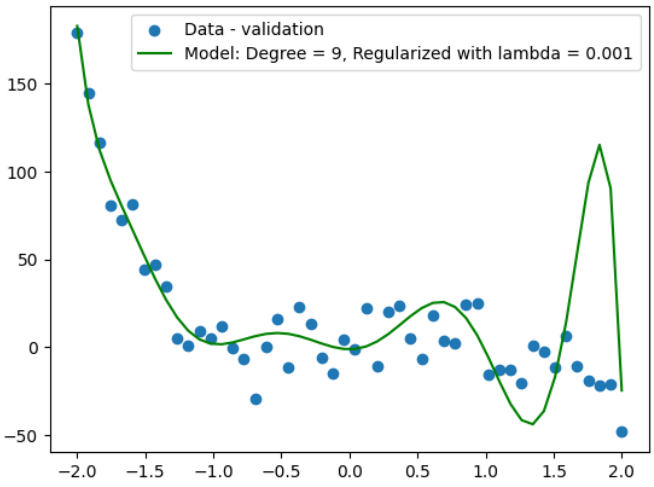

Figure 10: Validation data with $\lambda = 0.001$.

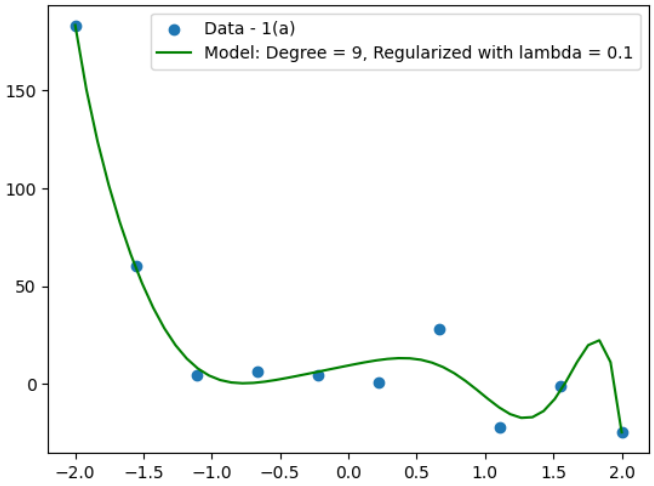

Figure 11: Regularizing with $\lambda = 0.1$ (Degree $= 9$).

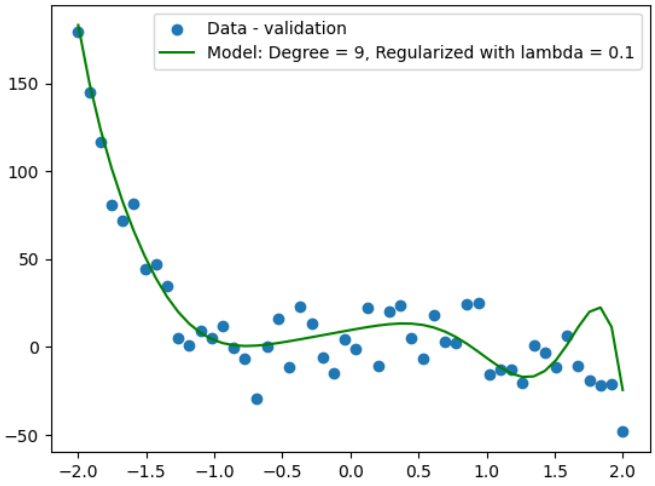

Figure 12: Validation data with $\lambda = 0.1$.

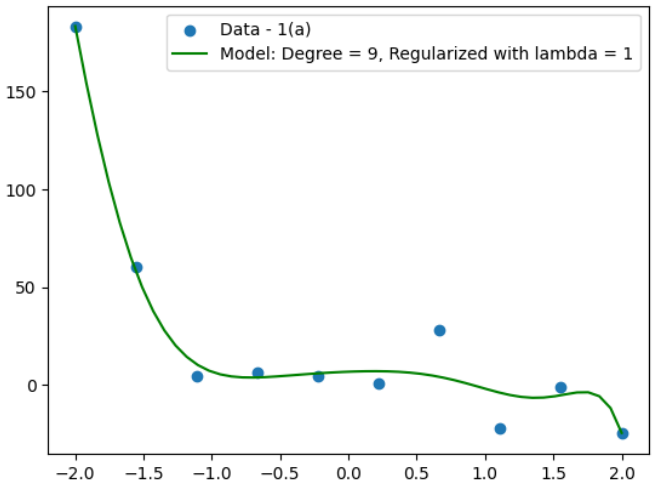

Figure 13: Regularizing with $\lambda = 1$ (Degree $= 9$).

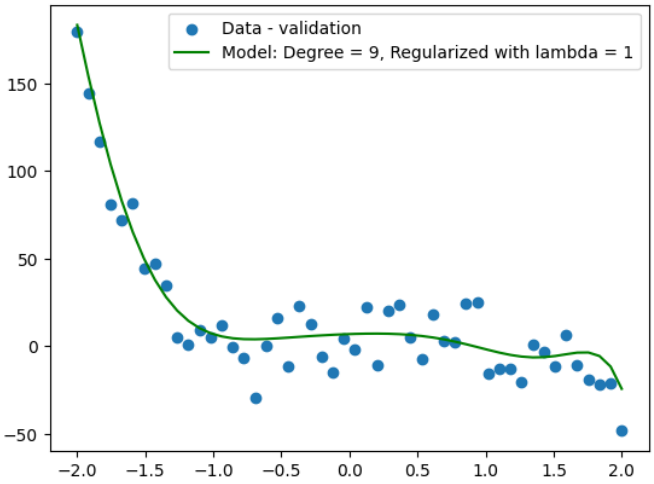

Figure 14: Validation data with $\lambda = 1$.

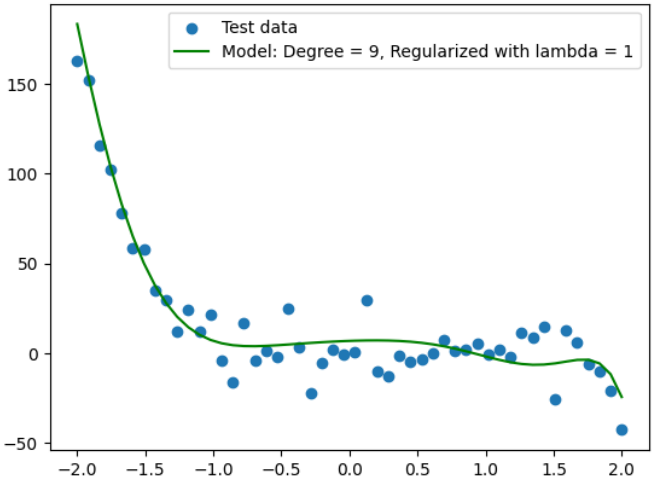

Figure 15: Predicting test data with the best regularized model

Dataset - 1(b): It turned out, there was no overfitting for models with degree $= 3, 5$ and 7, given an **overfitting ratio threshold of** 1.5, when compared with the **RMSE** from validation data. Following are the plots after applying regularization to models of degree 9, for different values of λ, from which we choose the best model.

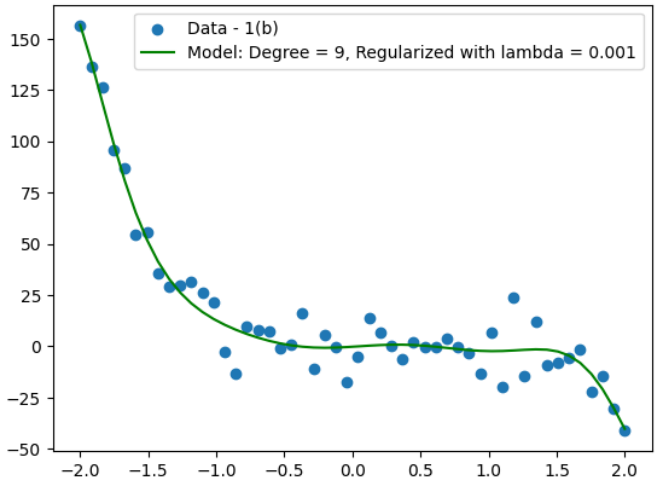

Figure 16: Regularizing with $\lambda = 0.001$ (Degree $= 9$).

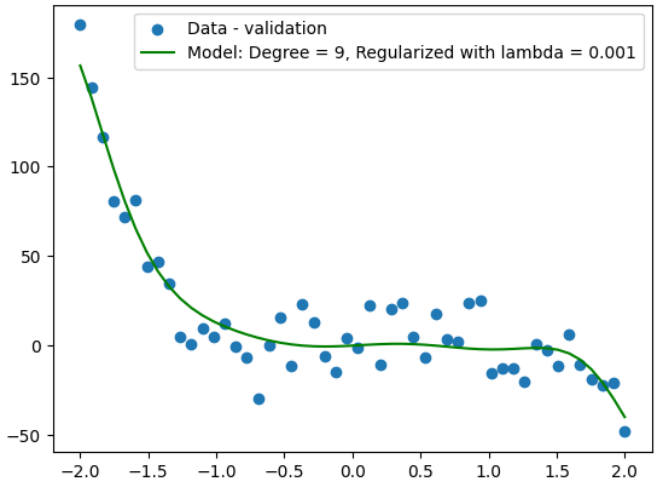

Figure 17: Validation data with $\lambda = 0.001$.

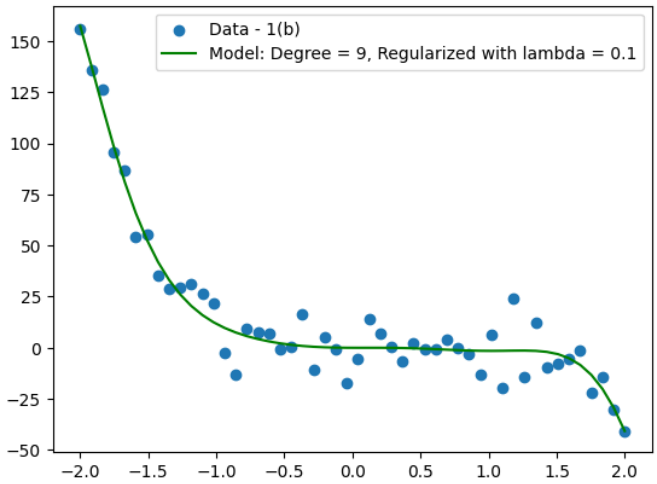

Figure 18: Regularizing with $\lambda = 0.1$ (Degree $= 9$).

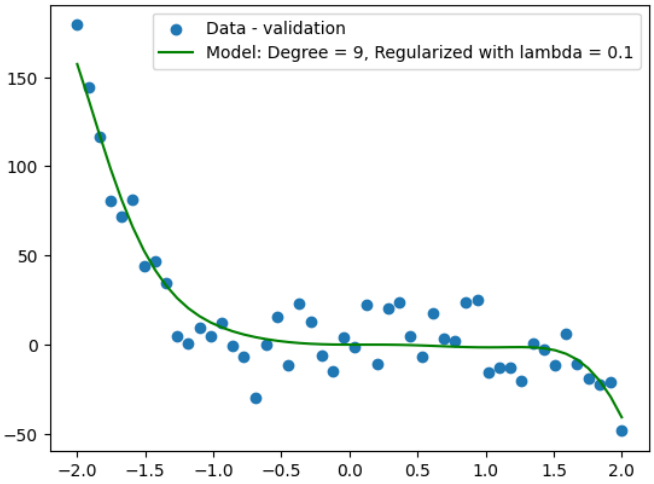

Figure 19: Validation data with $\lambda = 0.1$.

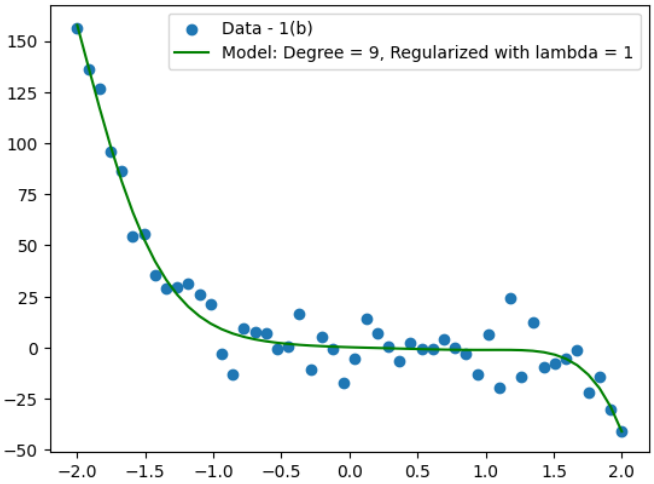

Figure 20: Regularizing with $\lambda = 1$ (Degree = 9).

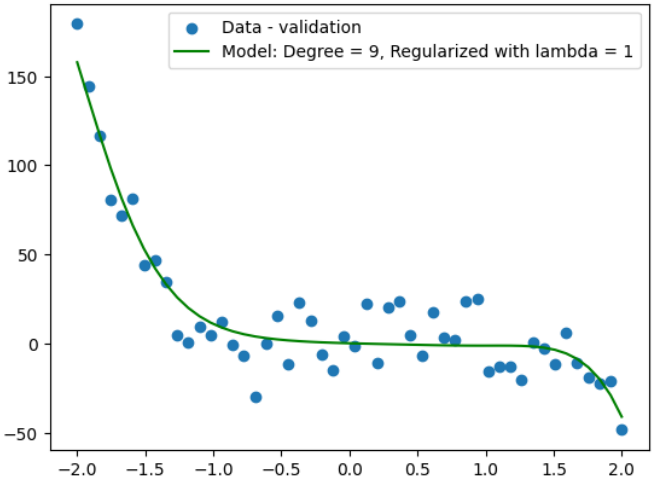

Figure 21: Validation data with $\lambda = 1$.

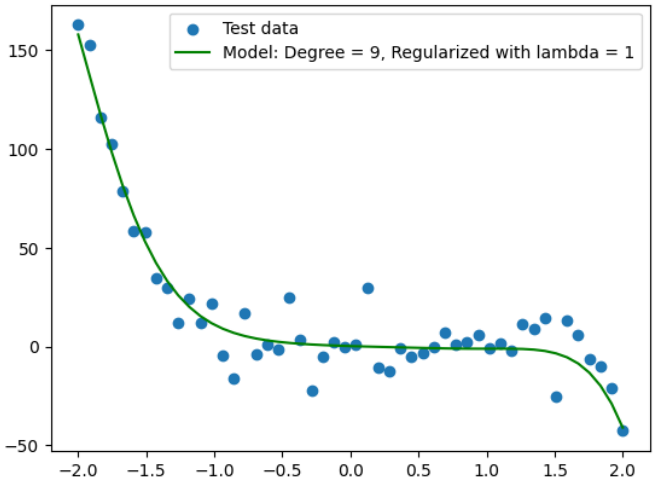

Figure 22: Predicting test data with the best regularized model

3.2.3 Results after Regularization

Table 2 summarizes the ERMS values for the training and validation sets at different polynomial degrees, and for different values of λ in the case of overfitting.

Table 2: ERMS values on Train and Validation data after regularization.

Degree, λ	**Train(1a)**	**Valid(1a)**	**Train(1b)**	**Valid(1b)**
3 (no overfit)	-	-	-	-
5 (no overfit)	-	-	-	-
7 (no overfit)	-	-	-	-
9, $\lambda = 0.001$	1.4632	34.6557	9.0117	13.7974
9, $\lambda = 0.1$	7.9913	16.0545	9.0356	13.7422
9, $\lambda = 1.0$	10.0993	13.4166	9.0658	13.6826

Clearly, we see that the best models for both datasets $1(a)$ and $1(b)$, after regularization, is achieved at **degree = 9**, and $\lambda = \mathbf{1.0}$.

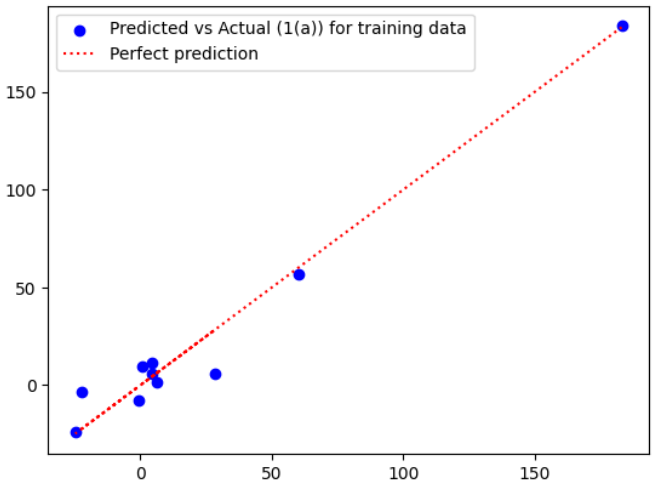

Figure 23: Predicted vs actual data (1(a))

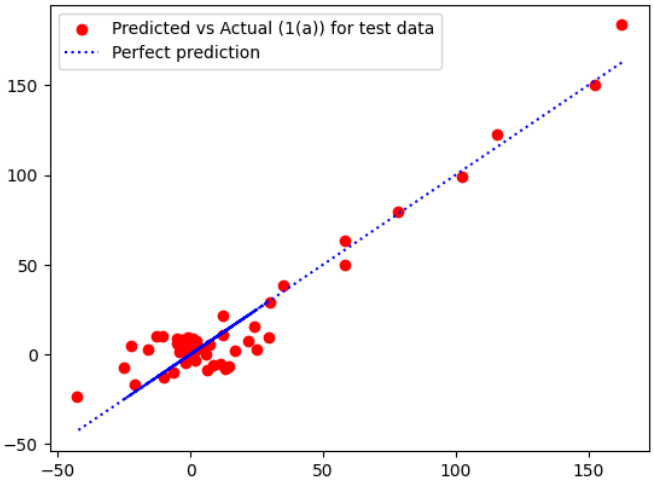

Figure 24: Predicted vs actual data on test data

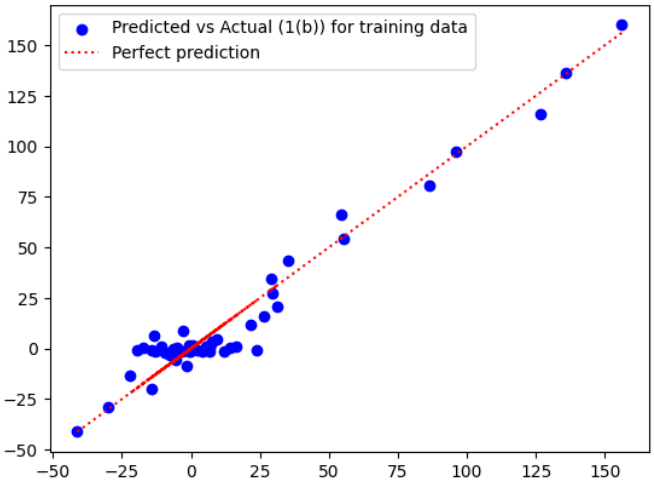

Figure 25: Predicted vs actual data (1(b))

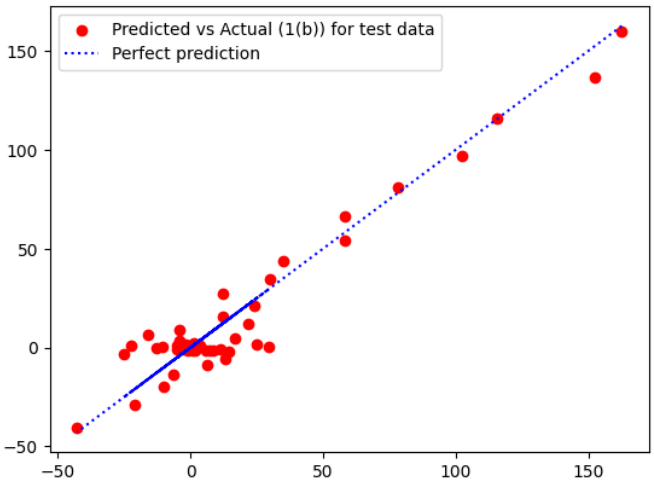

Figure 26: Predicted vs actual data on test data

3.3 Using Gaussian Basis Funtions

3.3.1 Results and Plots(Without Regularization)

Table 3 summarizes the ERMS values for the training and validation sets at different polynomial degrees.

Table 3: ERMS values on Train and Validation data (without regularization).

Best (k, σ)	Train ERMS	Validation ERMS
(5, 2.0)	11.5085 (1(a))	13.4332 (1(a))
(10, 3.0)	9.1163 (1(b))	13.6954(1(b))

The following figures show Gaussian Regression results for the best values of k and σ (the corresponding best values of the hyper-parameters k and σ are obtained by grid-search). Each plot compares the training data against the predicted Gaussian curve.

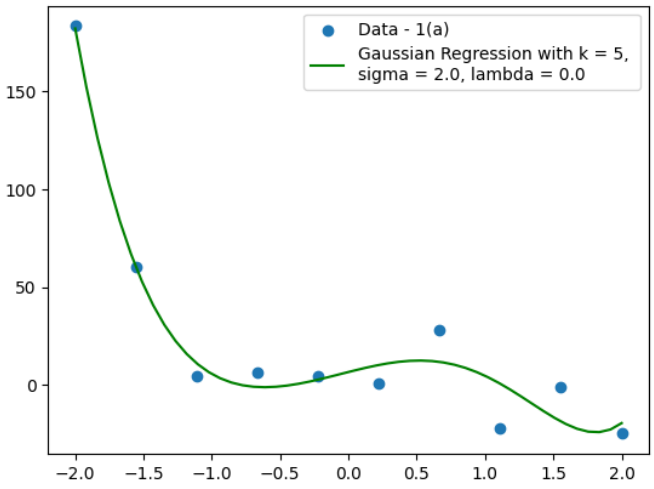

Figure 27: Gaussian on 1(a) with no regularization

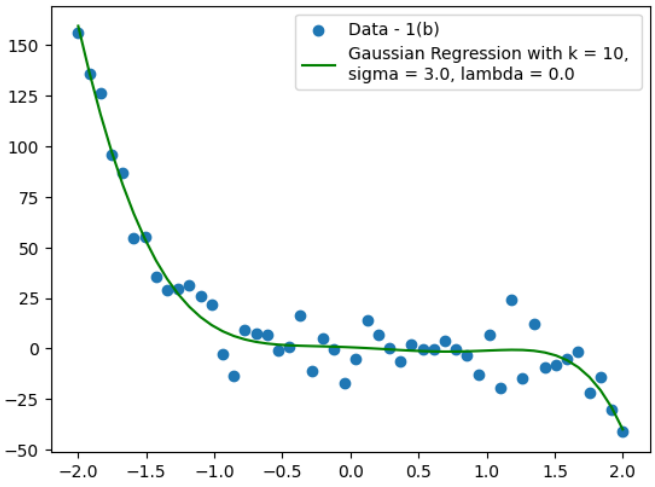

Figure 28: Gaussian on 1(b) with no regularization

3.3.2 Applying Regularization

Dataset - 1(a): We fix an overfitting threshold of **1.25**, and regularize the overfit models (without regularization), with the given values of λ.

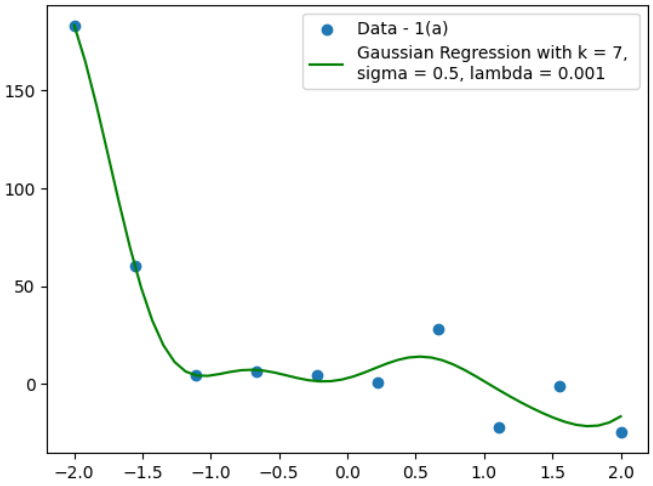

Figure 29: Gaussian on 1(a) with $k = 7, \sigma = 0.5, \lambda = 0.001$

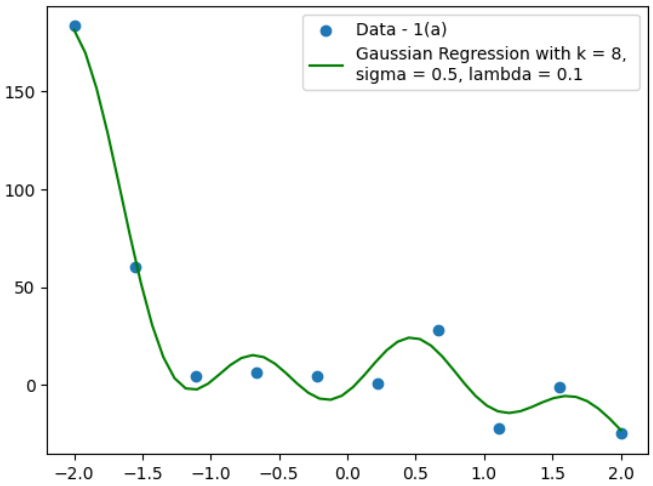

Figure 30: Gaussian on 1(a) with $k = 8, \sigma = 0.5, \lambda = 0.1$

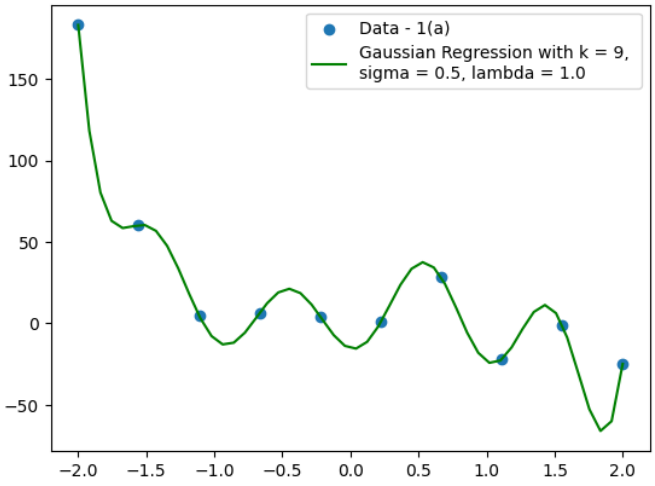

Figure 31: Gaussian on 1(a) with $k = 9, \sigma = 0.5, \lambda = 1.0$

Dataset - 1(b): We again fix the same overfitting threshold of **1.25**, and regularize the overfit models (without regularization), with the given values of λ.

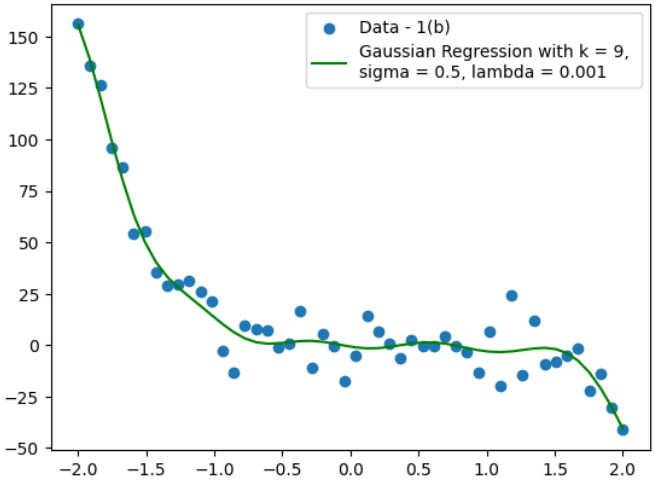

Figure 32: Gaussian on 1(b) with $k = 9, \sigma = 0.5, \lambda = 0.001$

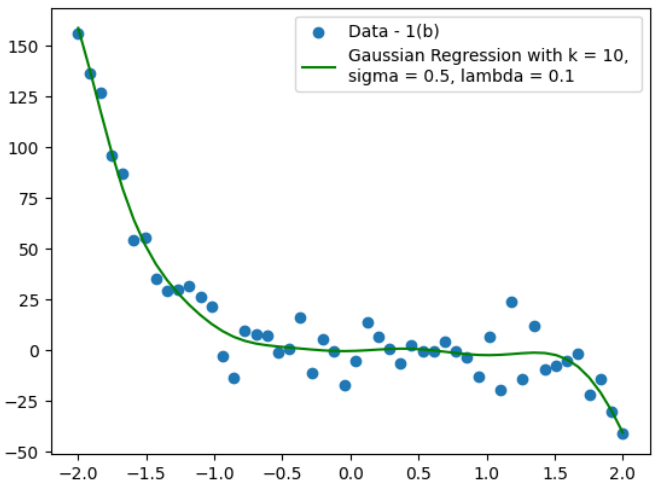

Figure 33: Gaussian on 1(b) with $k = 10, \sigma = 0.5, \lambda = 0.1$

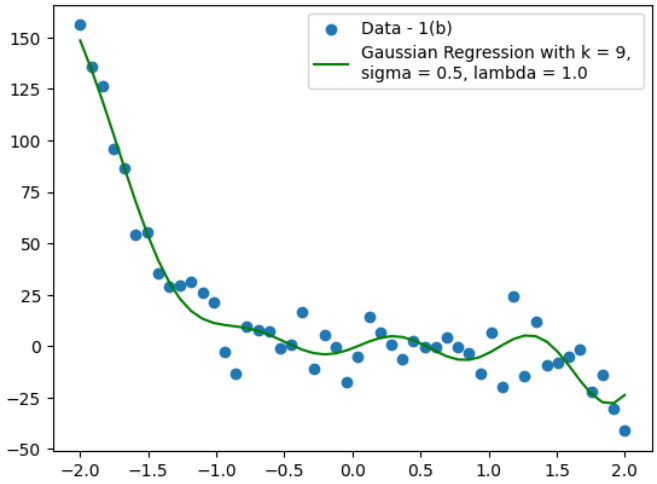

Figure 34: Gaussian on 1(b) with $k = 9, \sigma = 0.5, \lambda = 1.0$

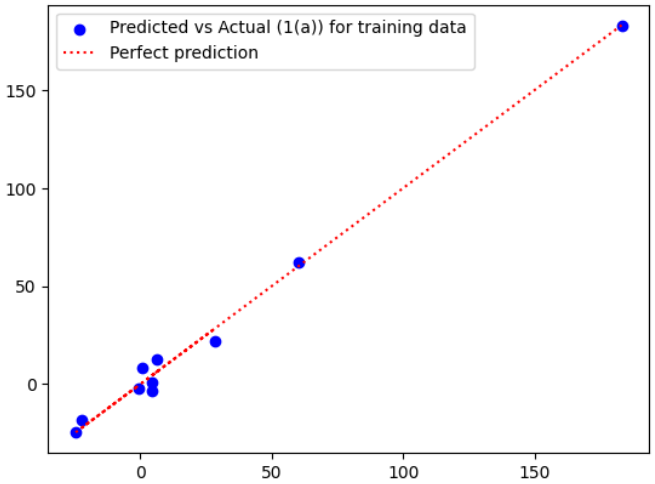

Figure 35: Predicted vs actual data (1(a))

3.3.3 Results after Regularization

Table 4 and 5 summarizes the ERMS values for the training and validation sets for different values of (k, σ, λ)

Table 4: ERMS values on Train and Validation data (1(a)) (after regularization).

(k, σ, λ)	**Train(1(a))**	**Valid(1(a))**
(7, 0.5, 0.001)	9.4291	15.0540
(8, 0.5, 0.1)	10.7326	15.8858
(9, 0.5, 1.0)	29.8113	21.6825

Table 5: ERMS values on Train and Validation data (1(b)) (after regularization).

(k, σ, λ)	**Train(1(b))**	**Valid(1(b))**
(9, 0.5, 0.001)	16.2734	16.4710
(10, 0.5, 0.1)	14.9050	16.1689
(9, 0.5, 1.0)	17.4218	21.0985

On investigating across all models (both with and without regularization), we find that the best model for both datasets 1(a) and 1(b) is $(k = 7, \sigma = 1.0, \lambda = 0.0)$, i.e., an unregularized model.

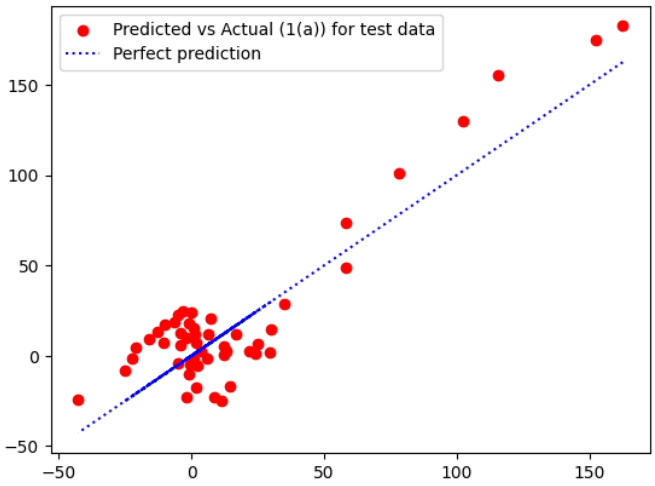

Figure 36: Predicted vs actual data on test data

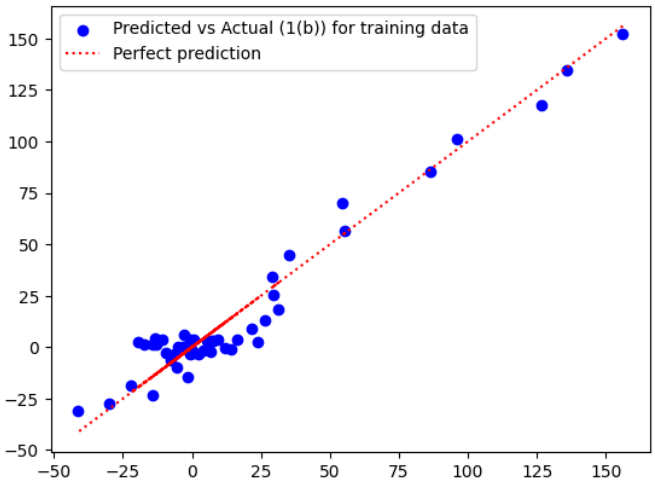

Figure 37: Predicted vs actual data (1(b))

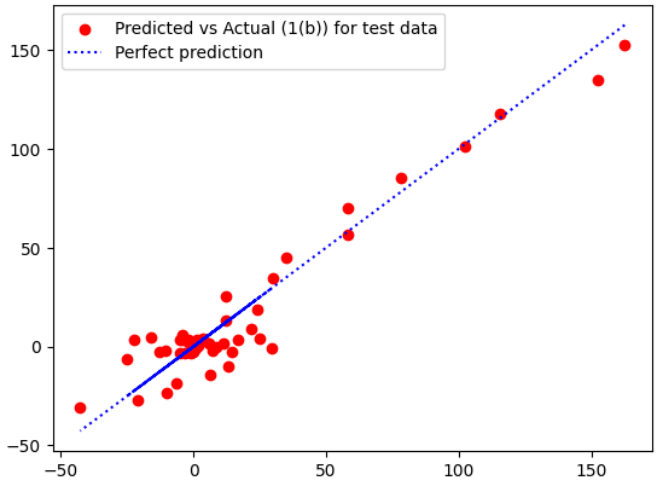

Figure 38: Predicted vs actual data on test data

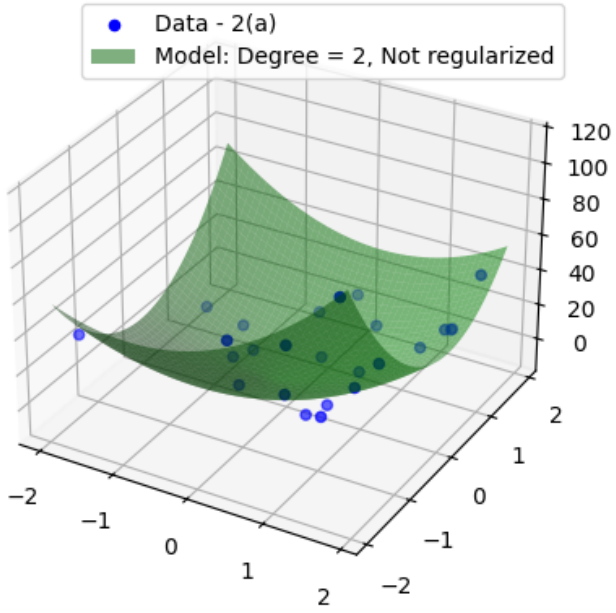

Figure 39: Polynomial Curve Fitting 2(a) (Degree = 2).

4 Dataset2

4.1 Description

Input data is 2-dimensional (Bivariate).

1. Training Dataset 2(a): 25 examples

2. Training Dataset 2(b): 100 examples

Training Dataset 2(a) and 2(b)

4.2 Using Polynomial Basis Functions

4.2.1 Results and Plots(Without Regularization)

Table 6 summarizes the ERMS values for the training and validation sets at different polynomial degrees.

Table 6: ERMS values on Train and Validation data (without regularization).

Degree	Train-2a	Valid-2a	Train-2b	Valid-2b
2	10.0720	11.4945	23.1709	24.5344
4	1.4766	5.3649	23.0079	24.5017
6	4.5712	11.7588	22.8417	24.5286
8	1.2689	13.6942	22.8248	24.5244

The following figures show polynomial curve fitting results for different degrees. Each plot compares the training data against the fitted polynomial curve.

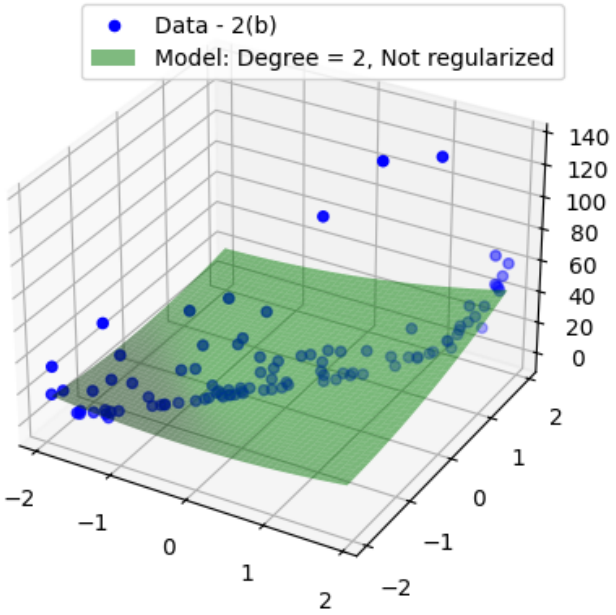

Figure 40: Polynomial Curve Fitting 2(b) (Degree = 2).

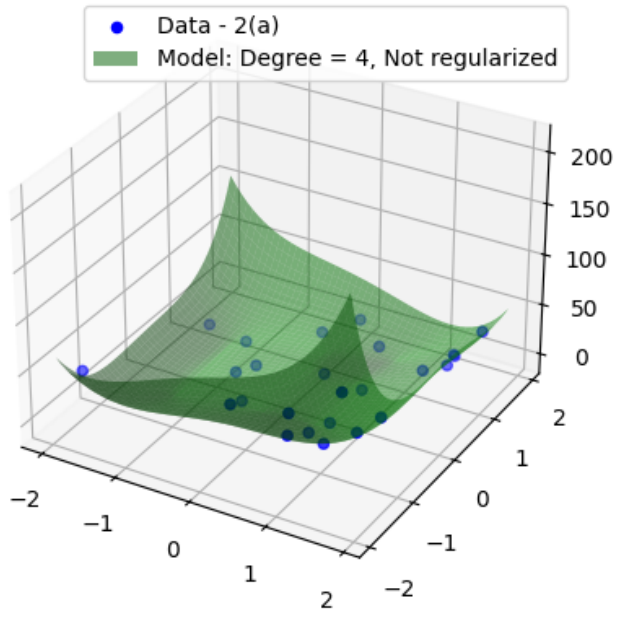

Figure 41: Polynomial Curve Fitting 2(a) (Degree = 4).

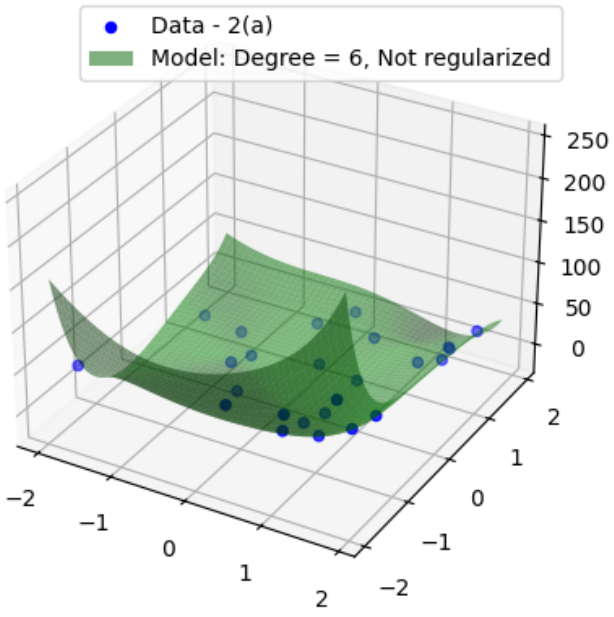

Figure 43: Polynomial Curve Fitting 2(a) (Degree = 6).

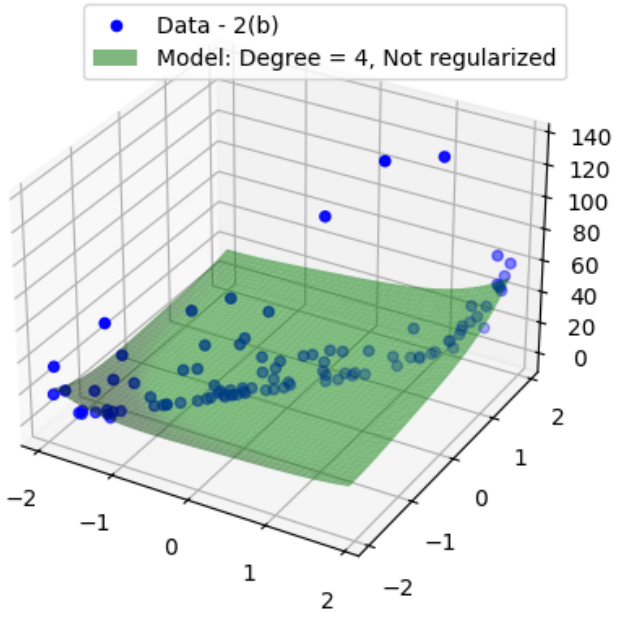

Figure 42: Polynomial Curve Fitting 2(b) (Degree = 4).

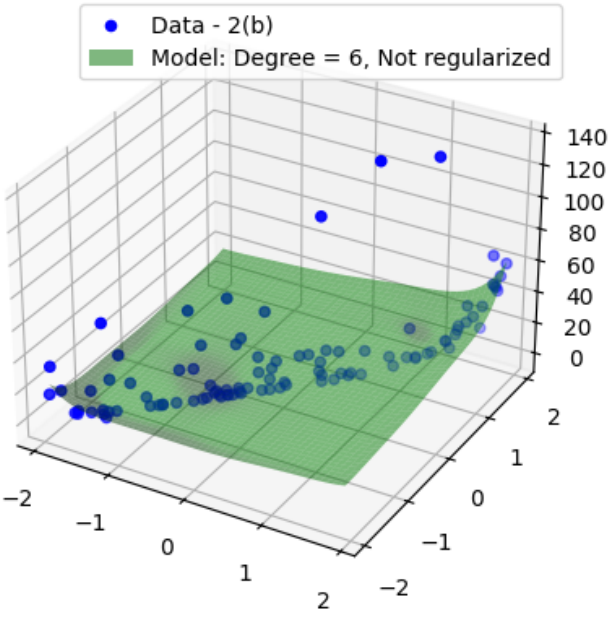

Figure 44: Polynomial Curve Fitting 2(b) (Degree = 6).

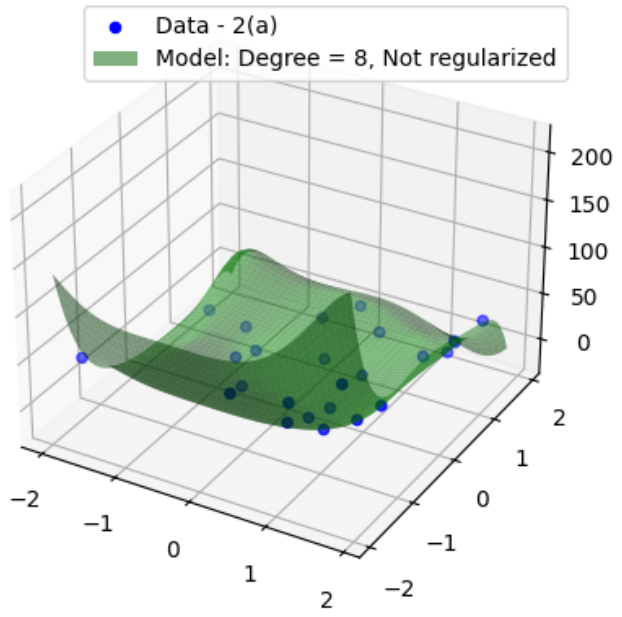

Figure 45: Polynomial Curve Fitting 2(a) (Degree = 8).

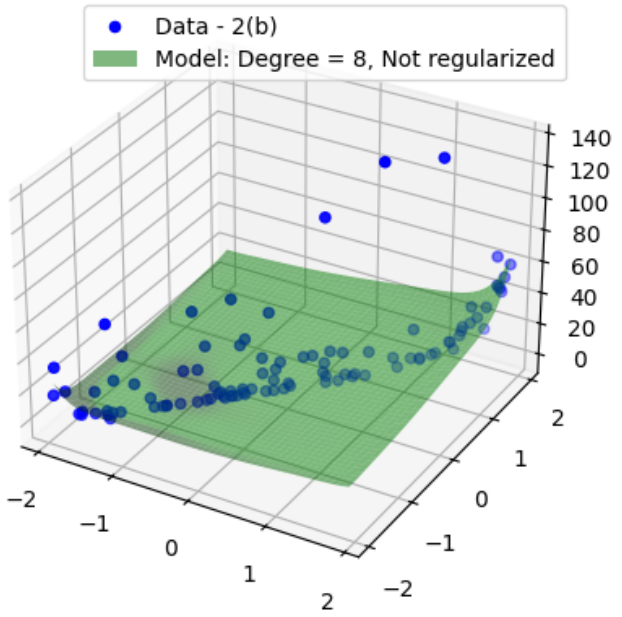

Figure 46: Polynomial Curve Fitting 2(b) (Degree = 8).

4.2.2 Applying Regularization

Dataset - 2(a): It turned out, there was no overfitting for models with degree = 2 and 4, given an **overfitting ratio threshold of** 1.5, when compared with the **RMSE** from validation data. Following are the plots after applying regularization to models of degree 6 and 8, for different values of λ, from which we choose the best model.

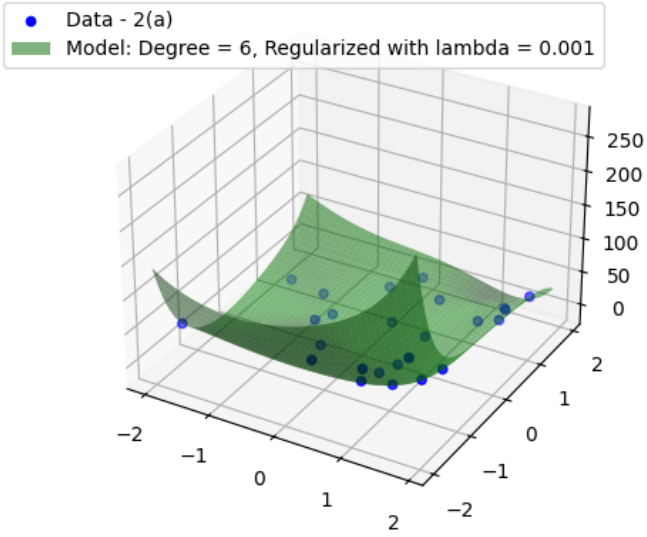

Figure 47: Regularizing with $\lambda = 0.001$ (Degree = 6).

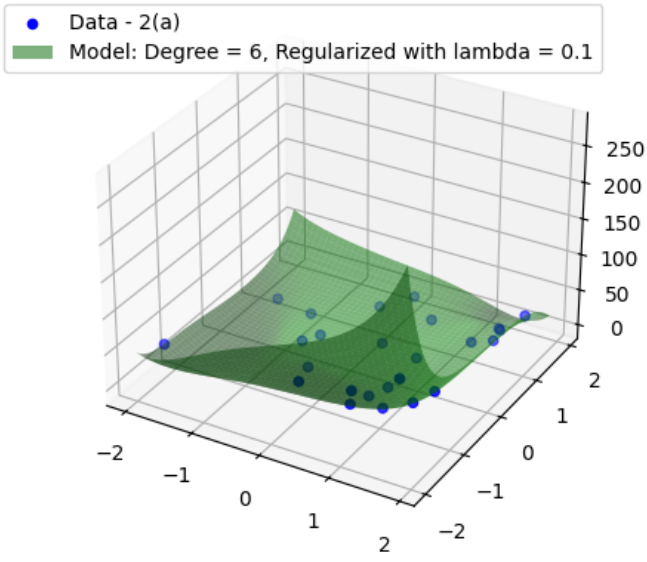

Figure 48: Regularizing with $\lambda = 0.1$ (Degree = 6).

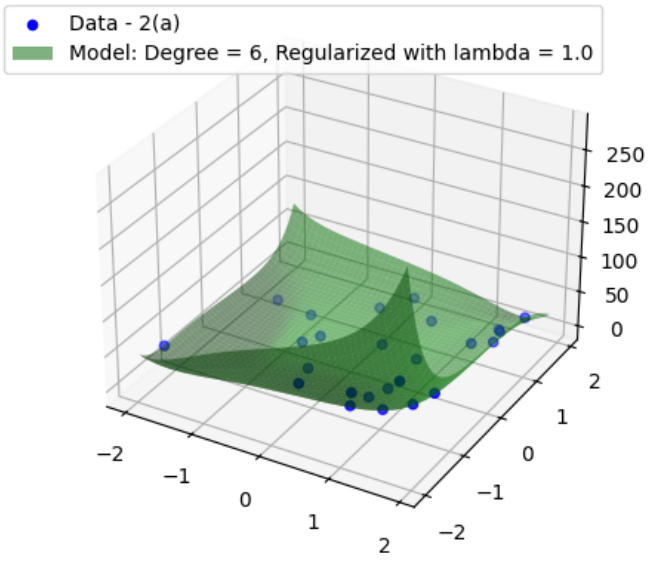

Figure 49: Regularizing with $\lambda = 1$ (Degree = 6).

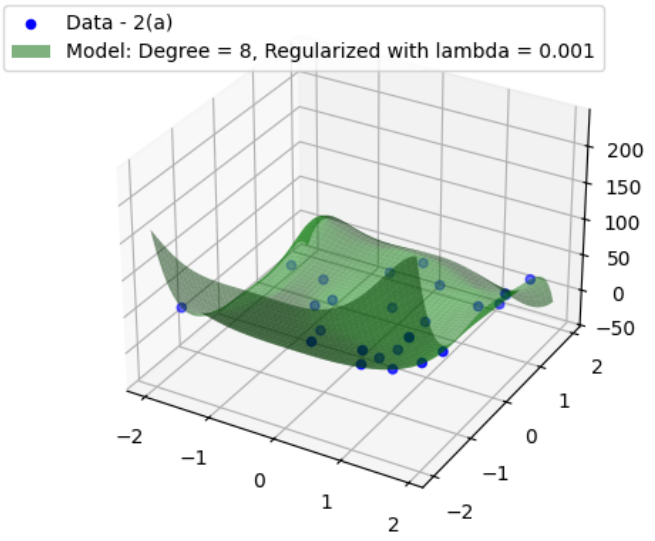

Figure 50: Regularizing with $\lambda = 0.001$ (Degree = 8).

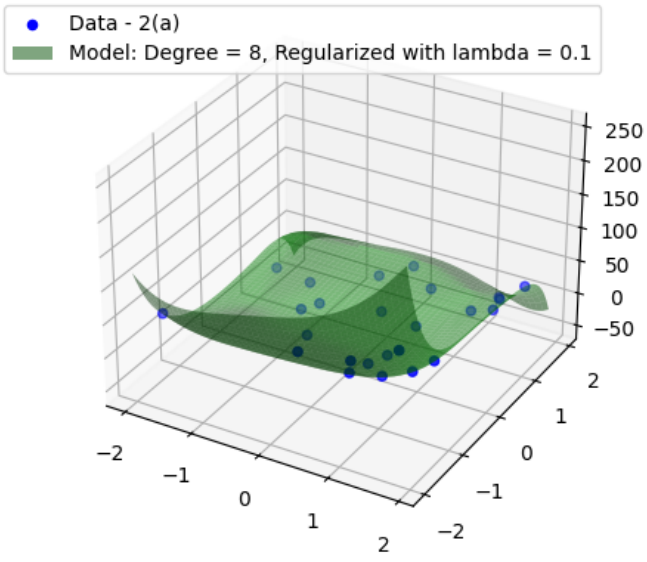

Figure 51: Regularizing with $\lambda = 0.1$ (Degree = 8).

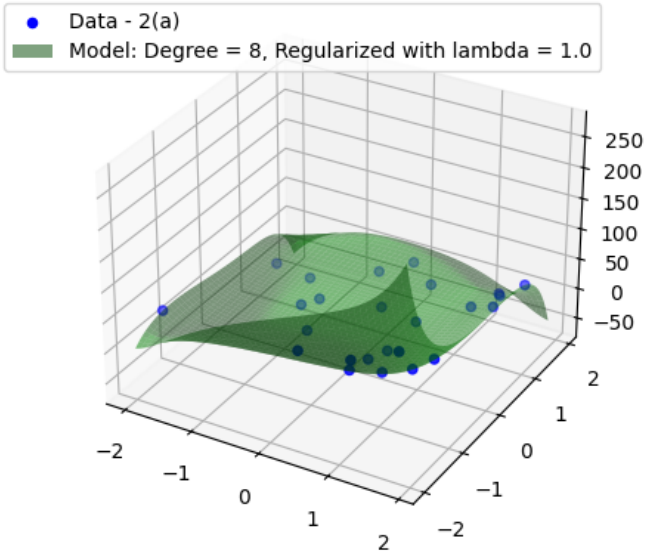

Figure 52: Regularizing with $\lambda = 1$ (Degree = 8).

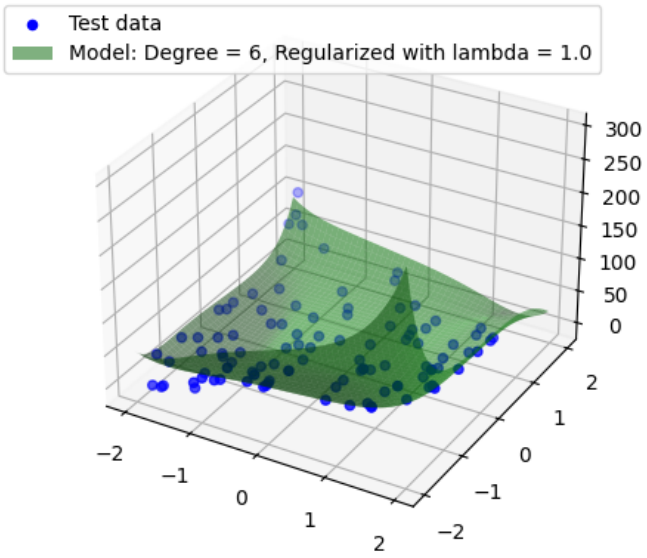

Figure 53: Predicting test data with the best regularized model

Dataset - 2(b): It turned out, there was no overfitting for models with degree = 2 and 4, given an **overfitting ratio threshold of** 1.07, when compared with the **RMSE** from validation data. Following are the plots after applying regularization to models of degrees 6 and 8, for different values of λ, from which we choose the best model.

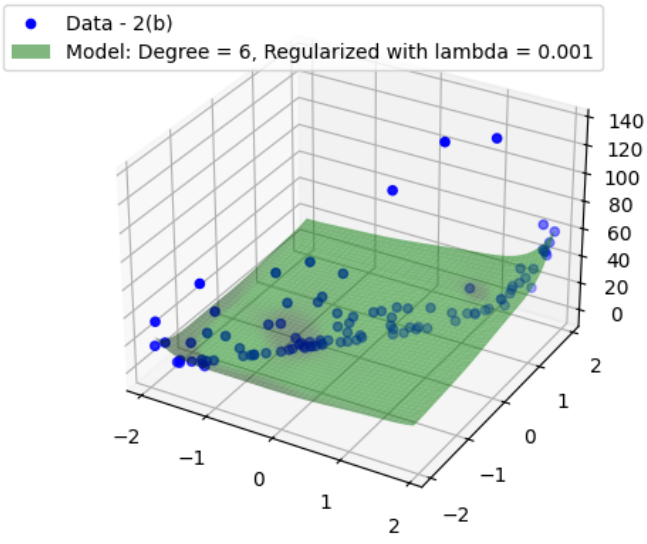

Figure 54: Regularizing with $\lambda = 0.001$ (Degree = 6).

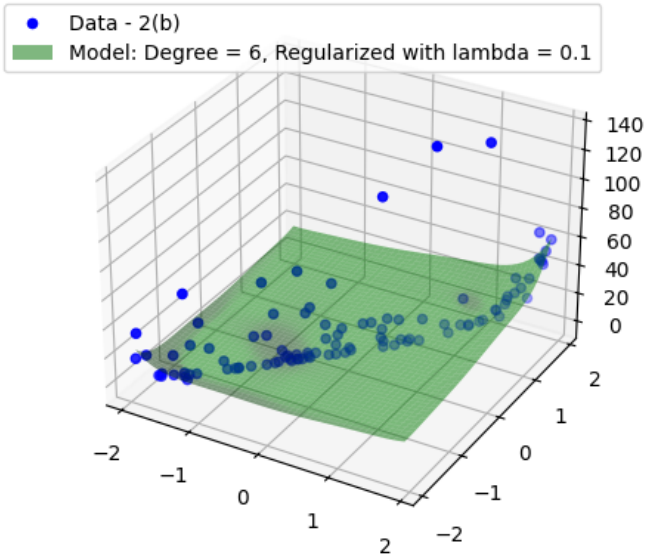

Figure 55: Regularizing with $\lambda = 0.1$ (Degree = 6).

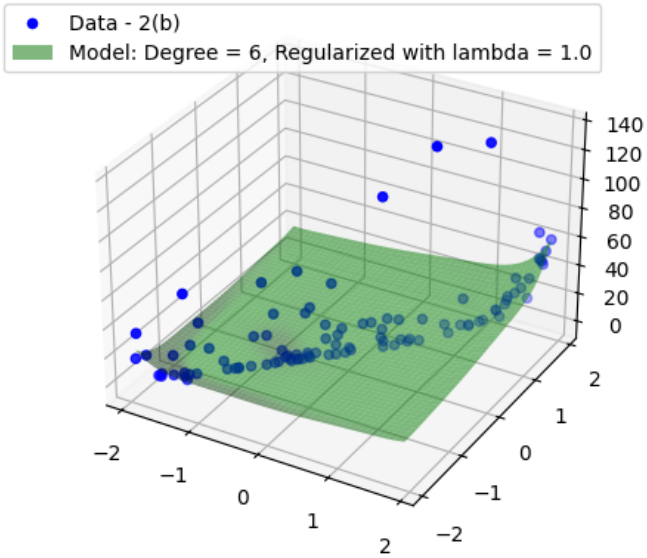

Figure 56: Regularizing with $\lambda = 1$ (Degree = 6).

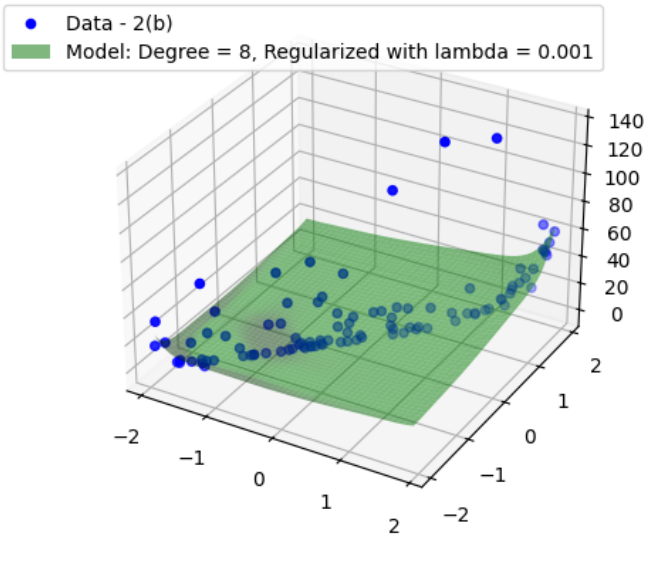

Figure 57: Regularizing with $\lambda = 0.001$ (Degree = 8).

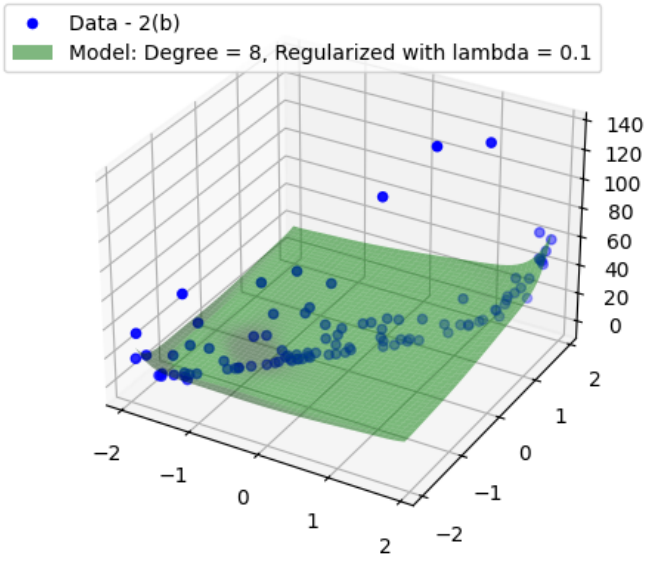

Figure 58: Regularizing with $\lambda = 0.1$ (Degree = 8).

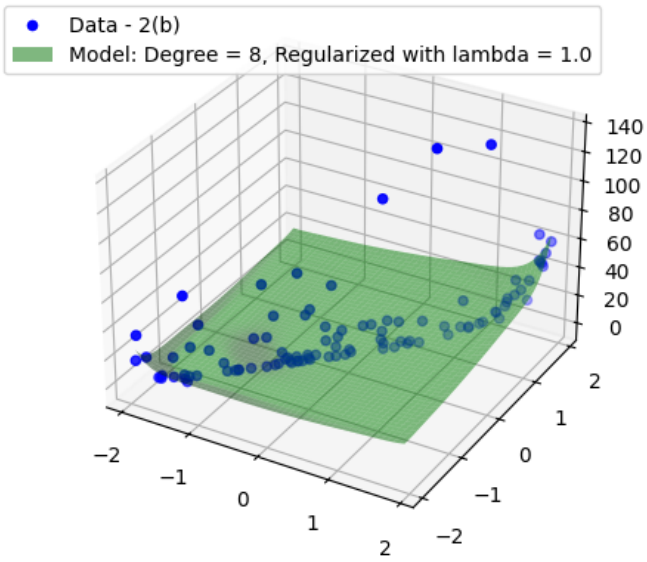

Figure 59: Regularizing with $\lambda = 1$ (Degree = 8).

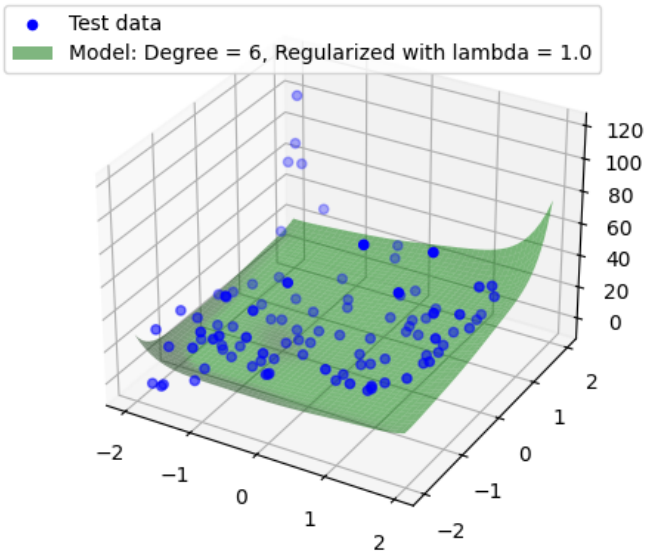

Figure 60: Predicting test data with the best regularized model

4.2.3 Results after Regularization

Table 7 summarizes the ERMS values for the training and validation sets at different polynomial degrees, and for different values of λ in the case of overfitting.

Table 7: ERMS values on Train and Validation data after regularization.

Degree, λ	**Tr(2a)**	**Val(2a)**	**Tr(2b)**	**Val(2b)**
2 (no overfit)	-	-	-	-
4 (no overfit)	-	-	-	-
6, $\lambda = 0.001$	0.0094	9.1657	22.8417	24.5286
6, $\lambda = 0.1$	0.1322	5.2447	22.8418	24.5263
6, $\lambda = 1.0$	0.3469	5.0942	22.8461	24.5099
8, $\lambda = 0.001$	0.0023	13.7901	22.8248	24.5244
8, $\lambda = 0.1$	0.0945	13.1604	22.8248	24.5245
8, $\lambda = 1.0$	0.2399	13.6470	22.8258	24.5238

Clearly, we see that the best models for both datasets $2(a)$ and $2(b)$, after regularization, is achieved at **degree = 6**, and $\lambda = \mathbf{1.0}$.

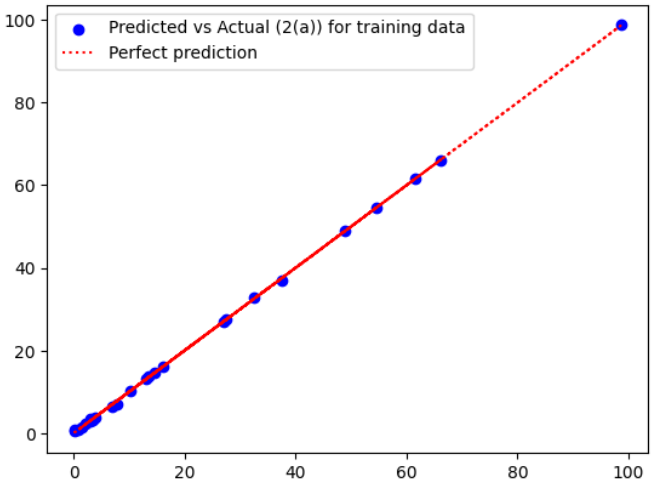

Figure 61: Predicted vs actual data (2(a))

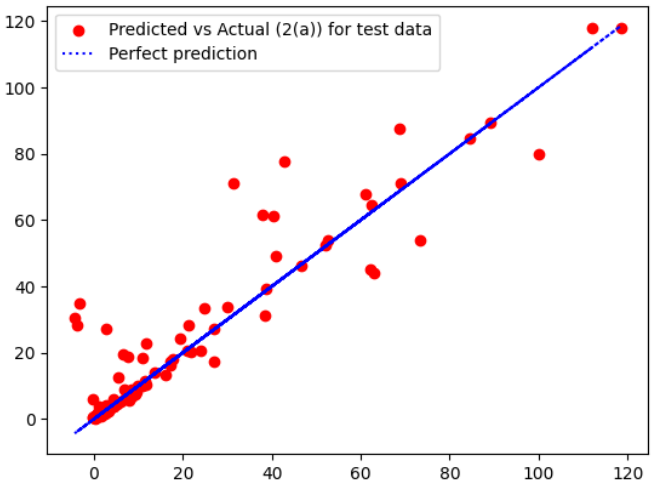

Figure 62: Predicted vs actual data on test data

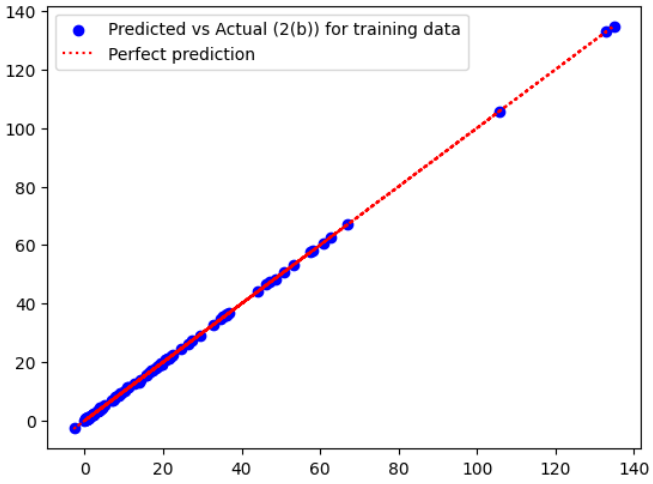

Figure 63: Predicted vs actual data (2(b))

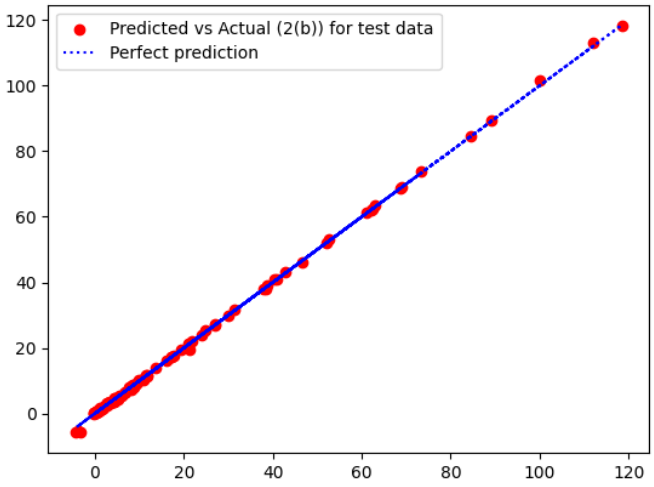

Figure 64: Predicted vs actual data on test data

4.3 Using Gaussian Basis Funtions

4.3.1 Results and Plots(Without Regularization)

Table 8 summarizes the ERMS values for the training and validation sets at different polynomial degrees.

Table 8: ERMS values on Train and Validation data (without regularization).

Best (k, σ)	**Train ERMS**	**Validation ERMS**
(4, 4.0)	10.6377 (2(a))	11.4035 (2(a))
(2, 2.0)	23.3476 (2(b))	22.3170 (2(b))

The following figures show Gaussian Regression results for the best values of k and σ (the corresponding best values of the hyper-parameters k and σ are obtained by grid-search). Each plot compares the training data against the predicted Gaussian curve.

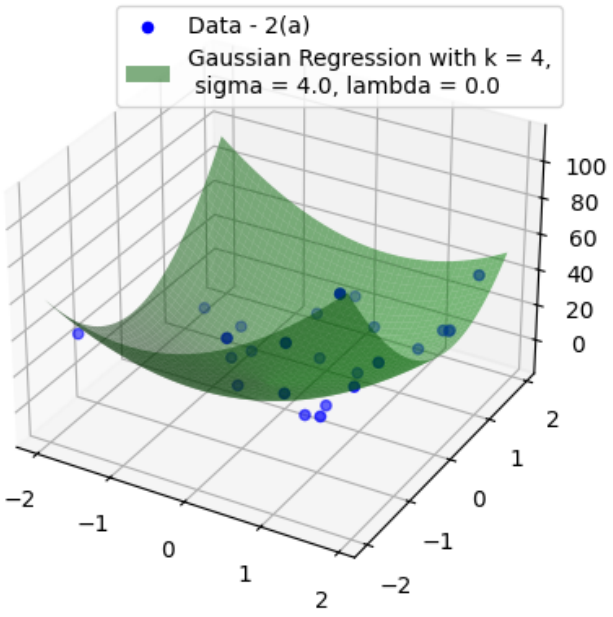

Figure 65: Gaussian on 2(a) with no regularization

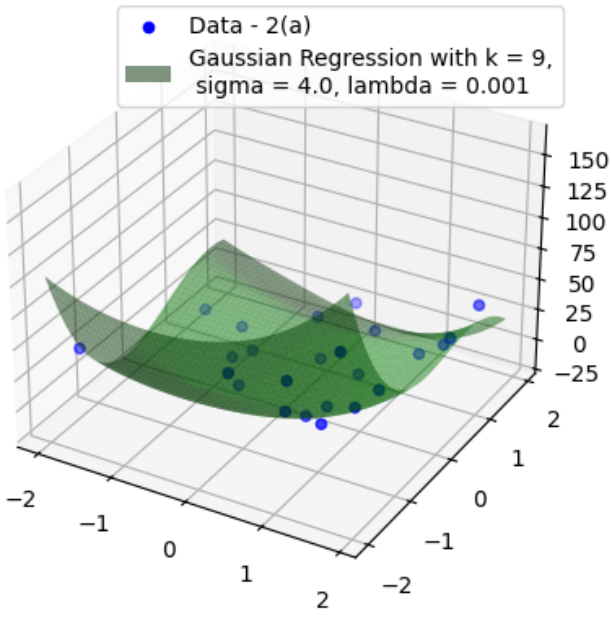

Figure 67: Gaussian on 1(a) with $k = 9, \sigma = 4.0, \lambda = 0.001$

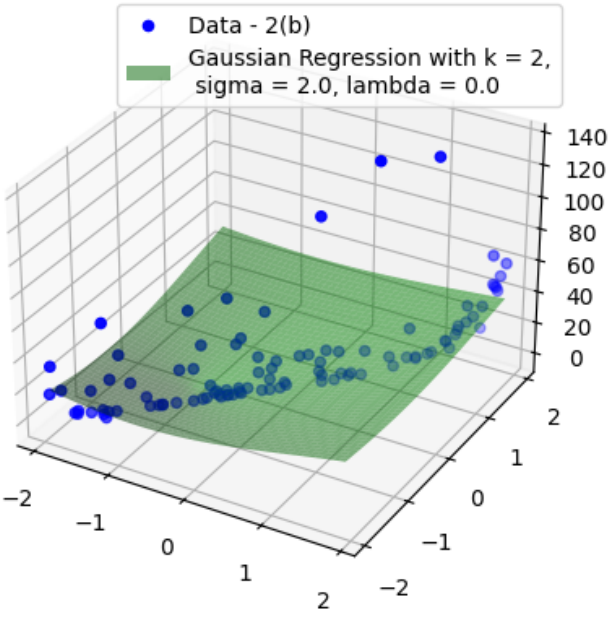

Figure 66: Gaussian on 2(b) with no regularization

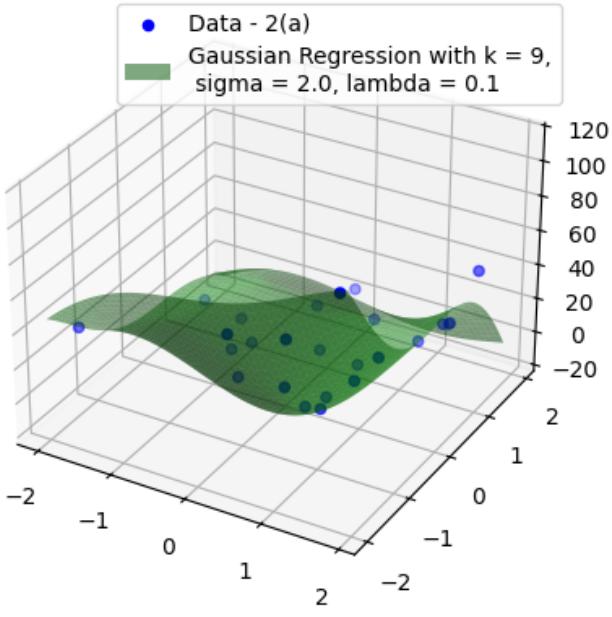

Figure 68: Gaussian on 1(a) with $k = 9, \sigma = 2.0, \lambda = 0.1$

4.3.2 Applying Regularization

Dataset - 2(a): We fix an overfitting threshold of **1.25**, and regularize the overfit models (without regularization), with the given values of λ.

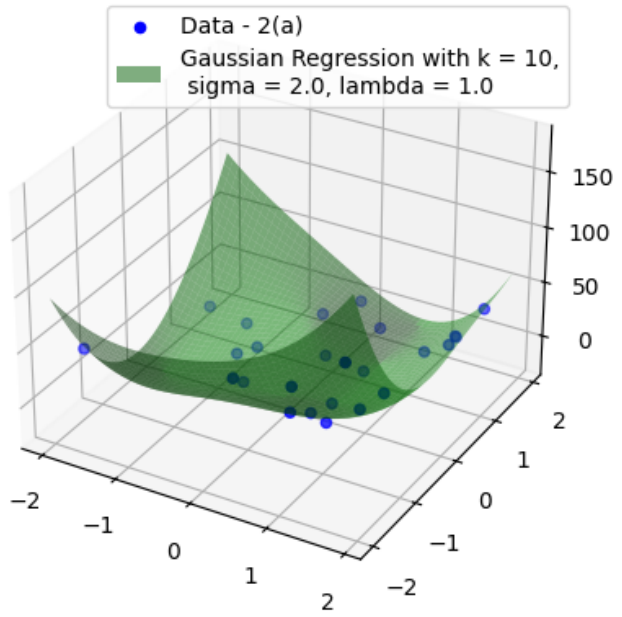

Figure 69: Gaussian on 1(a) with $k = 10, \sigma = 2.0, \lambda = 1.0$

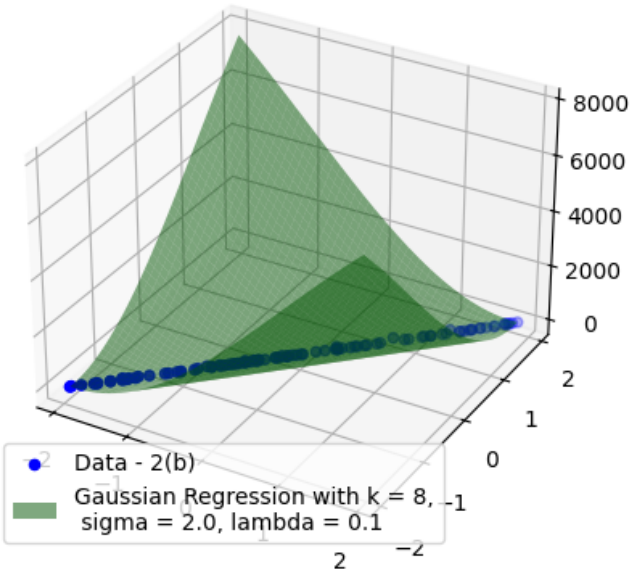

Figure 71: Gaussian on 2(b) with $k = 8, \sigma = 2.0, \lambda = 0.1$

Dataset - 2(b): We again fix the same overfitting threshold of **1.25**, and regularize the overfit models (without regularization), with the given values of λ.

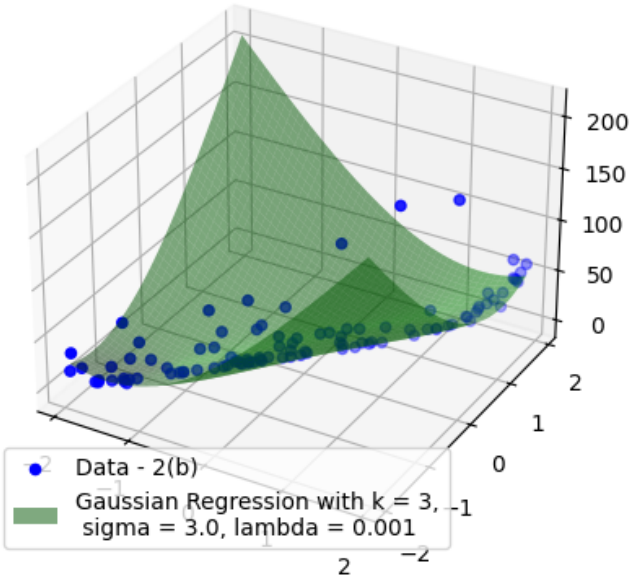

Figure 70: Gaussian on 2(b) with $k = 3, \sigma = 3.0, \lambda = 0.001$

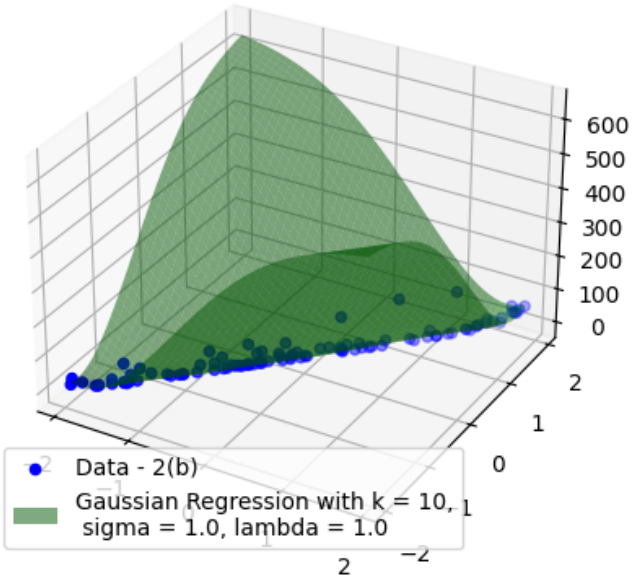

Figure 72: Gaussian on 2(b) with $k = 10, \sigma = 1.0, \lambda = 1.0$

4.3.3 Results after Regularization

Table 9 and 10 summarizes the ERMS values for the training and validation sets for different values of (k, σ, λ)

Table 9: ERMS values on Train and Validation data (2(a)) (after regularization).

(k, σ, λ)	**Train(2(a))**	**Valid(2(a))**
(9, 4.0, 0.001)	6.4924	14.9311
(9, 2.0, 0.1)	8.1093	18.4818
(10, 2.0, 1.0)	4.2123	12.0870

Table 10: ERMS values on Train and Validation data (2(b)) (after regularization).

(k, σ, λ)	**Train(2(b))**	**Valid(2(b))**
(3, 3.0, 0.001)	23.0569	50.7078
(8, 2.0, 0.1)	22.8246	92.5664
(10, 1.0, 1.0)	22.8150	98.7255

On investigating across all models (both with and without regularization), we find that the best model for both datasets 2(a) and 2(b) is $(k = 4, \sigma = 4.0, \lambda = 0.0)$, i.e., an unregularized model.

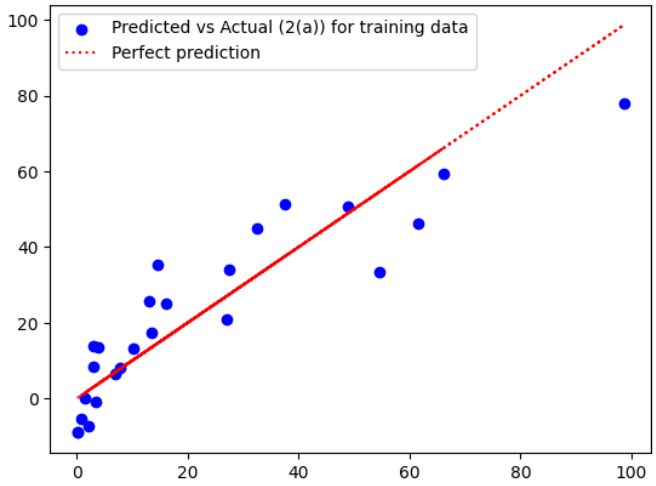

Figure 73: Predicted vs actual data (2(a))

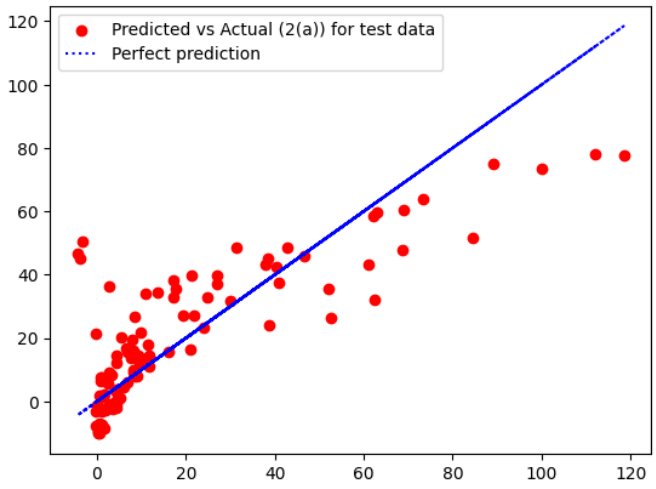

Figure 74: Predicted vs actual data on test data

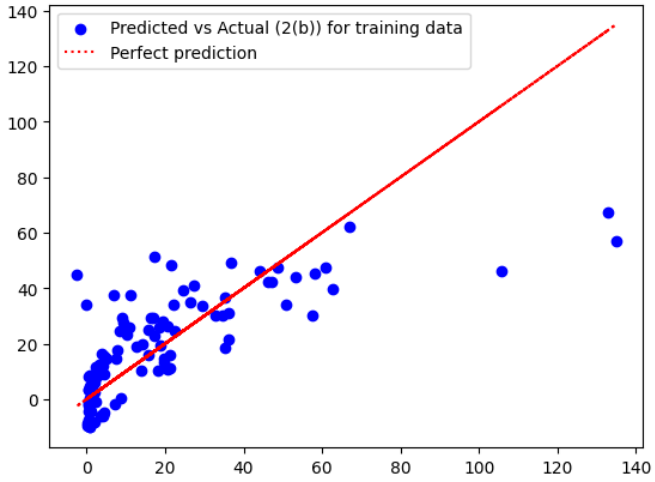

Figure 75: Predicted vs actual data (2(b))

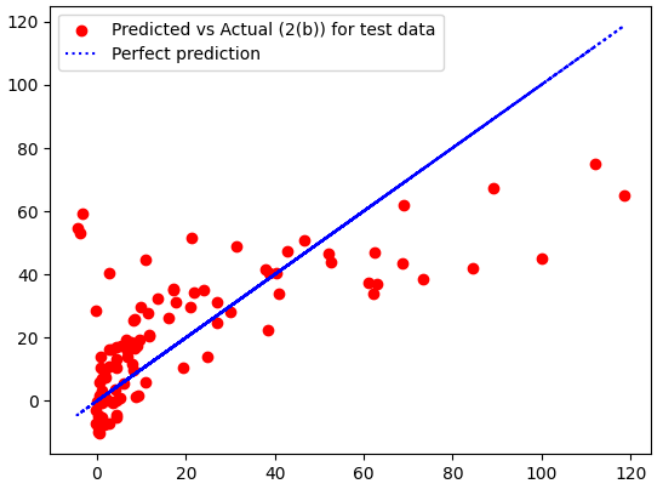

Figure 76: Predicted vs actual data on test data

5 Dataset3

5.1 Description

Input data is 3-dimensional (Trivariate). Output data is also 3-dimensional (Trivariate)

We model each output field as a **separate Linear Regression** problem.

1. Training Dataset 3: 350 examples

Training Dataset 3

5.2 Using Polynomial Basis Functions

5.2.1 Results and Plots(Without Regularization)

Table 11, 12, 13 summarizes the ERMS values for the training and validation (for y_1, y_2 and y_3, respectively) sets at different polynomial degrees, without regularization.

Table 11: ERMS values on Train and Validation data - y_1 (without regularization).

Degree	Training RMSE	Validation RMSE
2	0.4167	0.4431
3	0.4088	0.4478
4	0.3980	0.4639

Table 12: ERMS values on Train and Validation data - y_2 (without regularization).

Degree	Training RMSE	Validation RMSE
2	0.4066	0.3753
3	0.3968	0.3802
4	0.3857	0.3959

Table 13: ERMS values on Train and Validation data - y_3 (without regularization).

Degree	Training RMSE	Validation RMSE
2	0.4148	0.4213
3	0.4123	0.4159
4	0.4058	0.4332

5.2.2 Applying Regularization

Output - y_1: It turned out, there was no overfitting for models with degree = 2 and 3, given an **overfitting ratio threshold of** 1.15, when compared with the **RMSE** from validation data. Table 14 shows the results after applying regularization to models of degree 4, for different values of λ, from which we choose the best model.

Output - y_2: It turned out, there was no overfitting for any model, as it can be seen from Table 12 that, all Validation RMSEs are lower than Training RMSEs.

Output - y_3: It turned out, there was no overfitting for models with degree = 2 and 3, given an **overfitting ratio threshold of** 1.05, when compared with the **RMSE** from validation data. Table 16 shows the results after applying regularization to models of degree 4, for different values of λ, from which we choose the best model.

5.2.3 Results after Regularization

Tables 14, 15 and 16 summarizes the ERMS values for the training and validation sets at different polynomial degrees, and for different values of λ in the case of overfitting.

Table 14: ERMS values on Train and Validation data - y_1 (with regularization).

(Degree, λ)	Training RMSE	Validation RMSE
2 (No overfit)	-	-
3 (No overfit)	-	-
(4, 1e-6)	0.3980	0.4638
(4, 0.0001)	0.3984	0.4586
(4, 0.1)	0.4115	0.4415

Table 15: ERMS values on Train and Validation data - y_2 (with regularization).

(Degree, λ)	Training RMSE	Validation RMSE
2 (No overfit)	-	-
3 (No overfit)	-	-
4 (No overfit)	-	-

Table 16: ERMS values on Train and Validation data - y_3 (with regularization).

(Degree, λ)	Training RMSE	Validation RMSE
2 (No overfit)	-	-
3 (No overfit)	-	-
(4, 1e-6)	0.4058	0.4331
(4, 0.0001)	0.4060	0.4294
(4, 0.1)	0.4128	0.4201

Clearly, we see that the best models for fields y_1, y_2 and y_3 are achieved for (Degree = 4, $\lambda = 0.1$), (Degree = 2, $\lambda = 0.0$) and (Degree = 3, $\lambda = 0.0$), respectively.

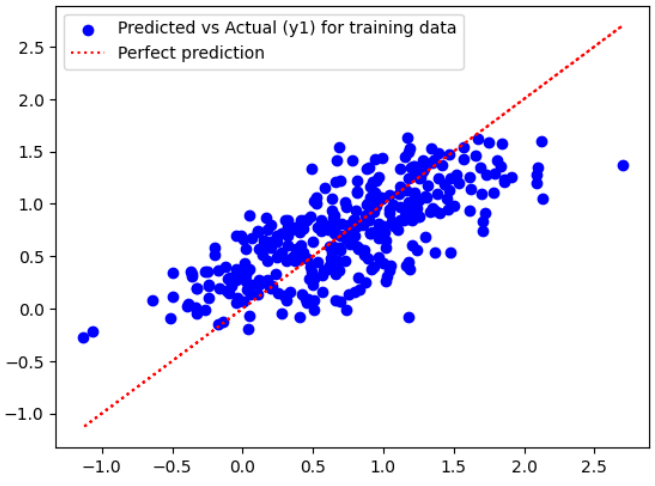

Figure 77: Predicted vs actual data (y_1)

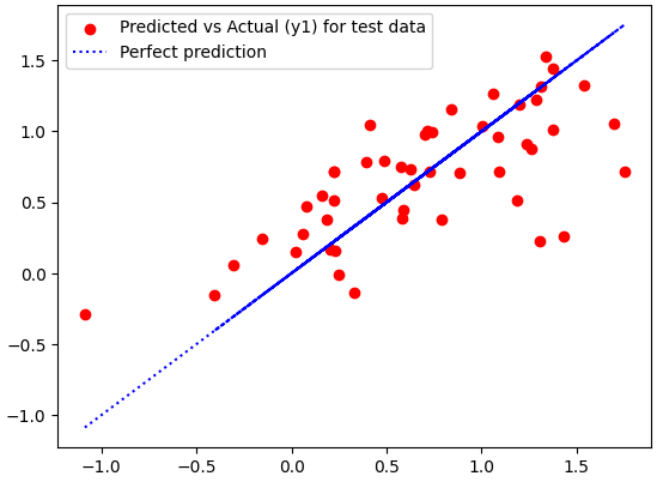

Figure 78: Predicted vs actual data on test data

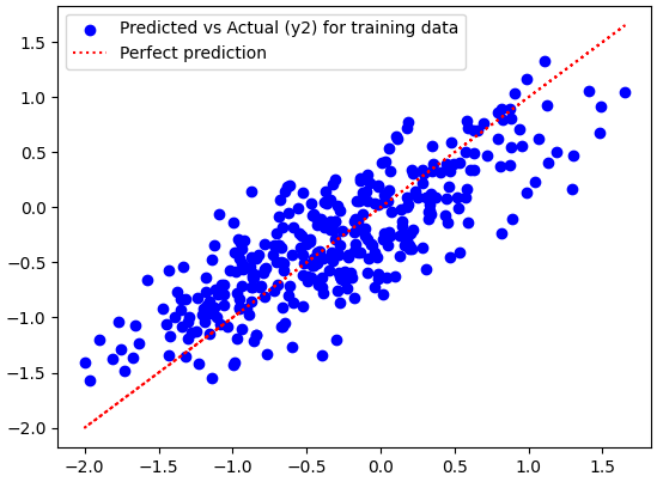

Figure 79: Predicted vs actual data (y_2)

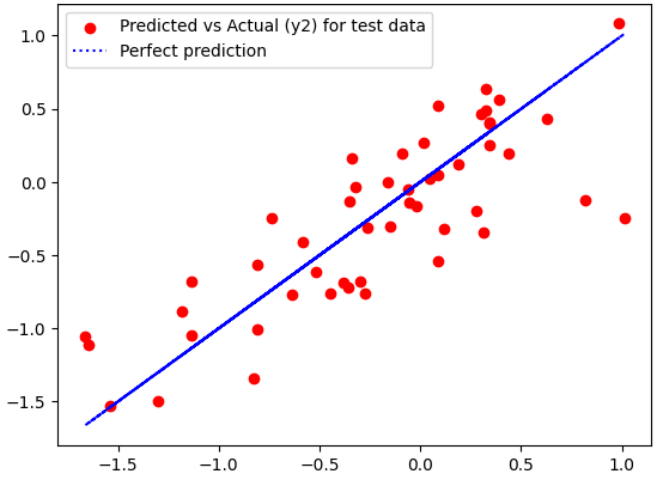

Figure 80: Predicted vs actual data on test data

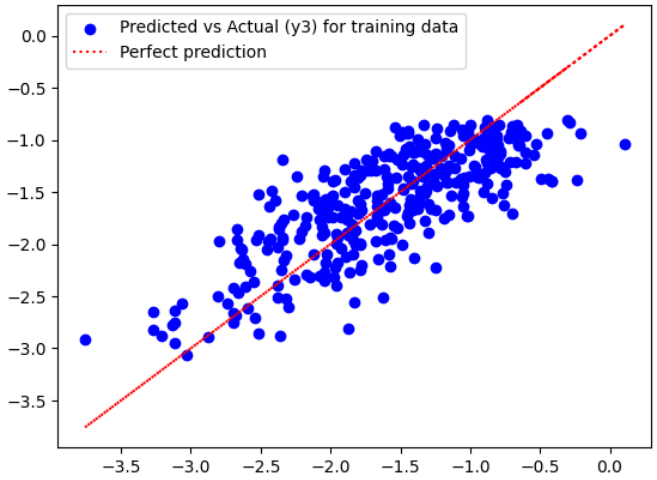

Figure 81: Predicted vs actual data (y_3)

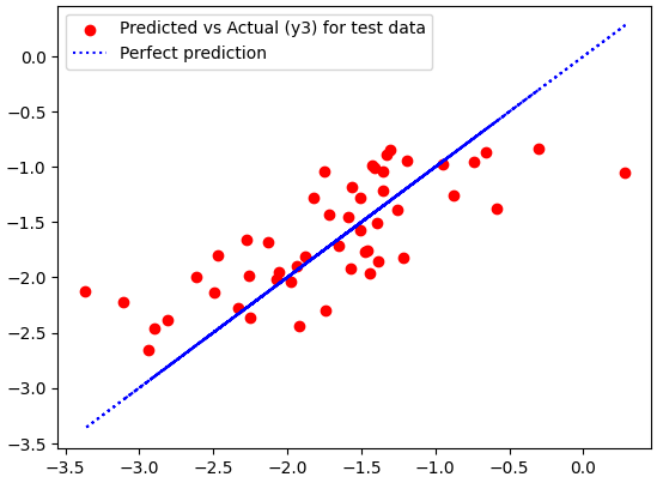

Figure 82: Predicted vs actual data on test data

5.3 Using Gaussian Basis Funtions

5.3.1 Results and Plots(Without Regularization)

Table 17, 18, 19 summarizes the ERMS values for the training and validation (for y_1, y_2 and y_3, respectively) sets at the best corresponding values of k and σ, without regularization.

Table 17: ERMS values on Train and Validation data - y_1 (without regularization).

(k, σ)	**Training RMSE**	**Validation RMSE**
(8, 5.0)	0.4217	0.4519

Table 18: ERMS values on Train and Validation data - y_2 (without regularization).

(k, σ)	**Training RMSE**	**Validation RMSE**
(8, 3.0)	0.4072	0.3744

Table 19: ERMS values on Train and Validation data - y_3 (without regularization).

(k, σ)	**Training RMSE**	**Validation RMSE**
(8, 0.5)	0.4222	0.4056

5.3.2 Applying Regularization

Output - y_1: It turned out, given an **overfitting ratio threshold of** 1.03, when compared with the **RMSE** from validation data, there is overfit. Table 20 shows the results after applying regularization to the models, for different values of λ, from which we choose the best model.

Output - y_2: It turned out, there was no overfitting for any model, as it can be seen from Table 18 that, all Validation RMSEs are lower than Training RMSEs.

Output - y_3: It turned out, there was no overfitting for any model, as it can be seen from Table 19 that, all Validation RMSEs are lower than Training RMSEs.

5.3.3 Results after Regularization

Tables 20, 21 and 22 summarizes the ERMS values for the training and validation sets at the best possible values of k and σ, corresponding to different values of λ in the case of overfitting.

Table 20: ERMS values on Train and Validation data - y_1 (with regularization).

(k, σ, λ)	**Training RMSE**	**Validation RMSE**
(8, 1.0, 1e-6)	0.4198	0.4474
(6, 1.0, 0.0001)	0.4264	0.4435
(5, 0.5, 0.1)	0.4310	0.4533

Table 21: ERMS values on Train and Validation data - y_2 (with regularization).

(k, σ, λ)	**Training RMSE**	**Validation RMSE**
(8, 3.0) (NA)	-	-

Table 22: ERMS values on Train and Validation data - y_3 (with regularization).

(Degree, λ)	**Training RMSE**	**Validation RMSE**
(8, 0.5) (NA)	-	-

Clearly, we see that the best models for fields y_1, y_2 and y_3 are achieved for $(k = 6, \sigma = 1.0, \lambda = 0.0001)$, $(k = 8, \sigma = 3.0, \lambda = 0.0)$ and $(k = 8, \sigma = 0.5, \lambda = 0.0)$, respectively.

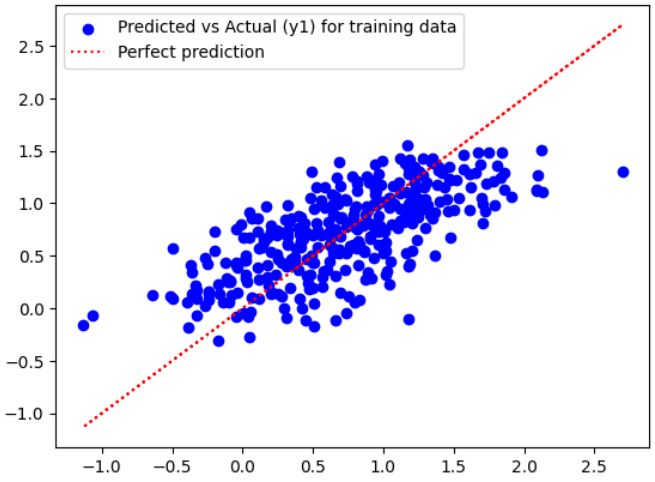

Figure 83: Predicted vs actual data (y_1)

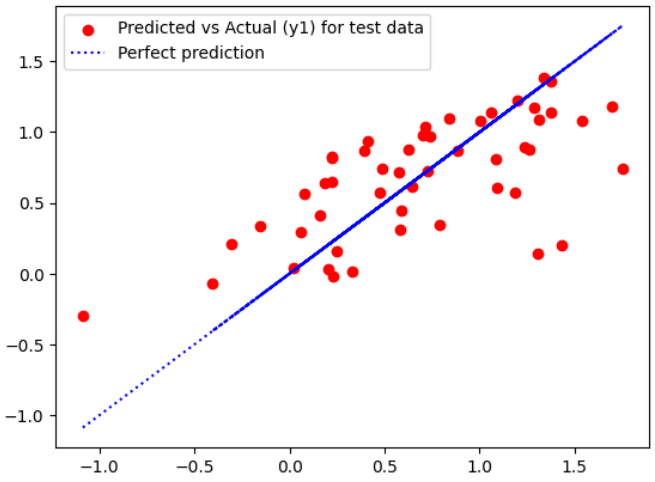

Figure 84: Predicted vs actual data on test data

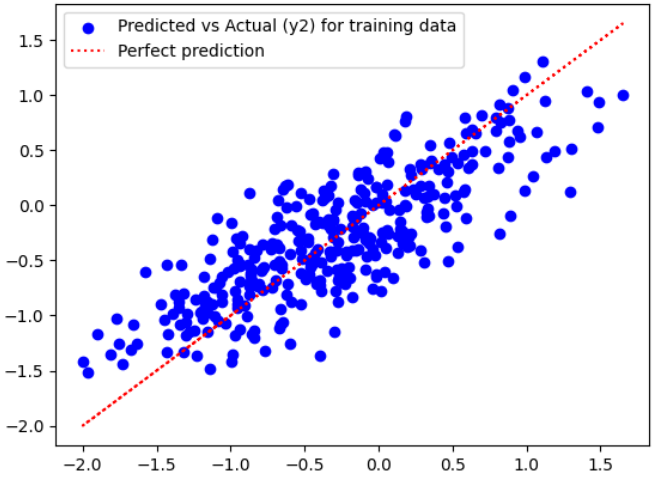

Figure 85: Predicted vs actual data (y_2)

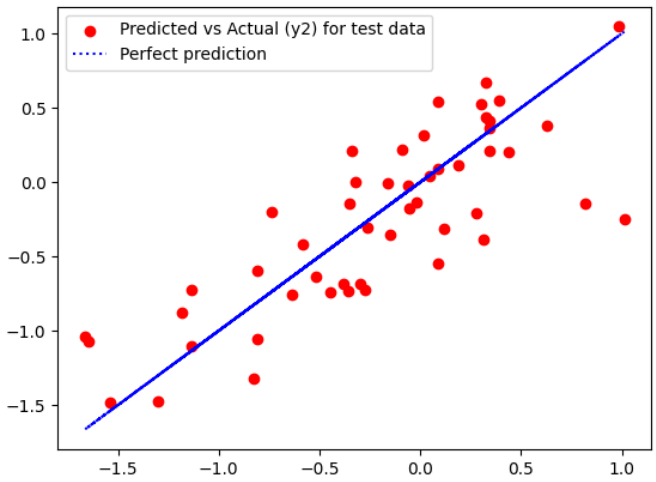

Figure 86: Predicted vs actual data on test data

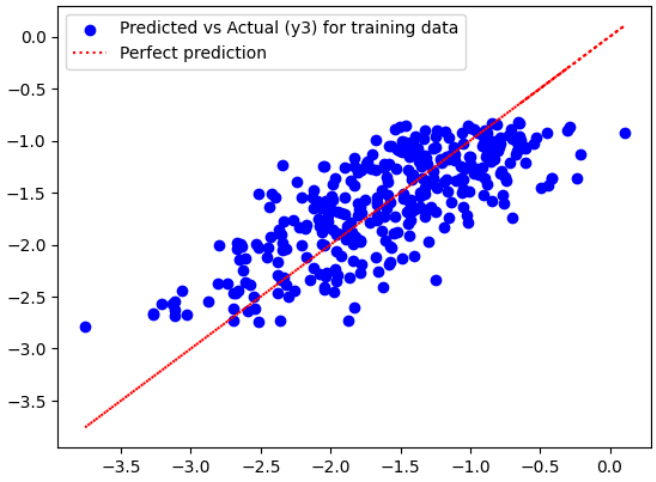

Figure 87: Predicted vs actual data (y_3)

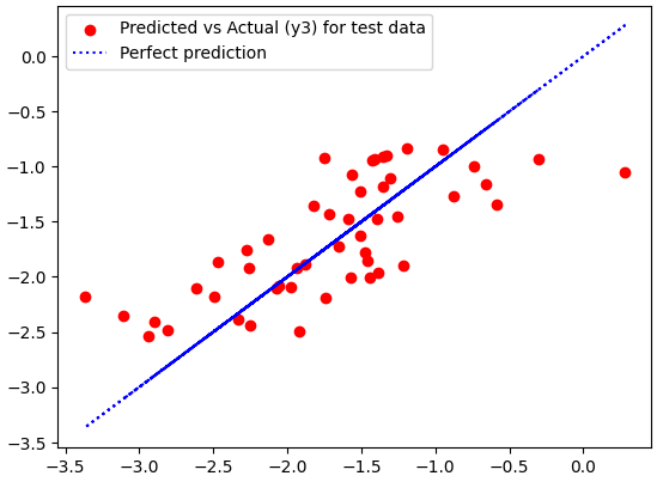

Figure 88: Predicted vs actual data on test data

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.