ASSIGNMENT 1

CS5691 Pattern Recognition and Machine Learning

CS5691 Assignemnt 1

Team Members:

BE17B007	N Sowmya Manojna
PH17B010	Thakkar Riya Anandbhai
PH17B011	Chaithanya Krishna Moorthy

Indian Institute of Technology, Madras

Contents

1 Task 1

1.1 Mathematical Formulation

The data for univariate polynomial regression is obtained by raising it to the required degree. In case of univariate polynomial regression of degree d, the dependent variable, of size $(d, 1)$ is assumed to have the form

$$\vec{y}_{n\times 1} = \phi_{n\times d} W_{d\times 1} \tag{1}$$

The weights corresponding to a given degree is then calculated by using the closed form solution for univariate polynomial regression:

$$W = (\phi^T \phi + \lambda I)^{-1} \phi^T \vec{y} \tag{2}$$

Where, λI is the regularization term.

1.2 Training and Validation Accuracies

In order to pick the parameters that best fit the dataset, a grid search was performed on the dataset. Prior to this, the dataset was split into training set, validation set and the testing set, in the ratio 70:10 (from the training data) :30. The results obtained is as follows:

Degree	λ	**Train Error**	**Validation Error**
6	0.0	0.044889	0.159636
3	0.0	0.672882	1.001484
9	0.5	0.750020	1.469413
2	0.0	1.014199	1.883134
9	1.0	1.040132	1.929033
9	2.0	1.354363	2.165779
9	10.0	2.281929	1.857270
9	50.0	3.342110	1.447933
9	100.0	3.782560	1.380623
9	0.0	5.063475	92.085167

Table 1: Results obtained for Task 1, with sample size of 10

Regularization was only applied in case of degree 9.

Degree	λ	**Train Error**	**Validation Error**
6	0.0	0.094536	0.094379
9	0.0	0.093581	0.100752
9	0.5	0.134226	0.152565
9	1.0	0.186479	0.209008
9	2.0	0.289107	0.311716
9	10.0	0.766298	0.776521

3	0.0	0.934079	0.862605
2	0.0	1.591842	1.421021
9	50.0	1.620063	1.707757
9	100.0	2.138200	2.310223

Table 2: Results obtained for Task 1, with sample size of 200

From the table above, we see that the best fit for the data is obtained for degree: 6 and $\lambda : 0$.

1.3 Model Fits

1.3.1 Sample Size: 10

The polynomial models and the corresponding fits obtained for sample size of 10 are as follows:

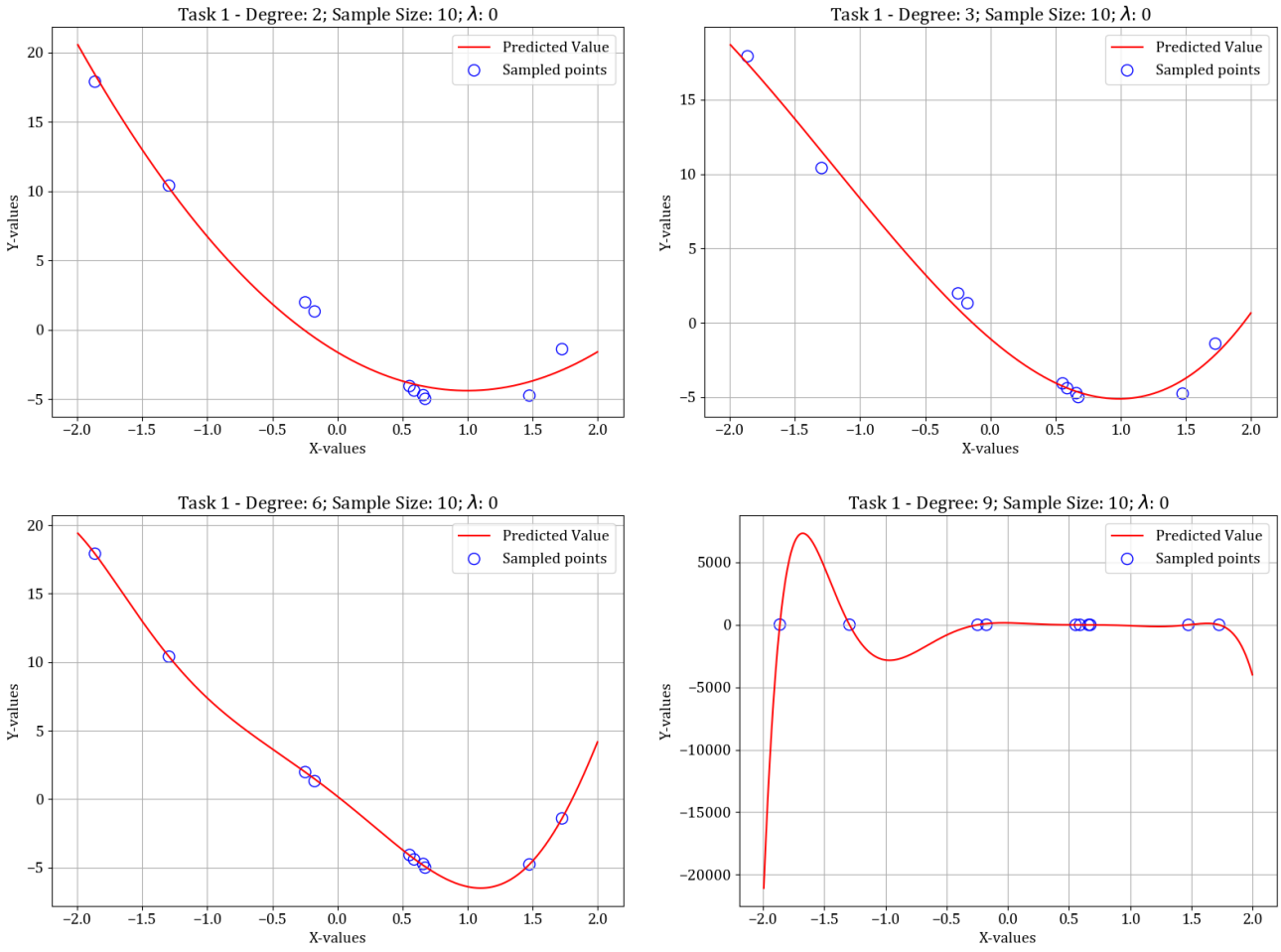


Figure 1: Task 1 - Polynomial fits, Sample size: 10

1.3.1.1 Inference

From the above plots, we can see that:

- Lower degree polynomial curves aren't able to model the dataset well (i.e.) the curve doesn't pass through all the data points.

- Higher degree polynomials are able to fit the dataset well. The curves pass through all the data points.
- However, the polynomial degree 9 curve seems to have a lot more variance along the y-axis than the remaining polynomial degrees.

1.3.2 Sample Size: 200

The polynomial models and the corresponding fits obtained for sample size of 200 are as follows:

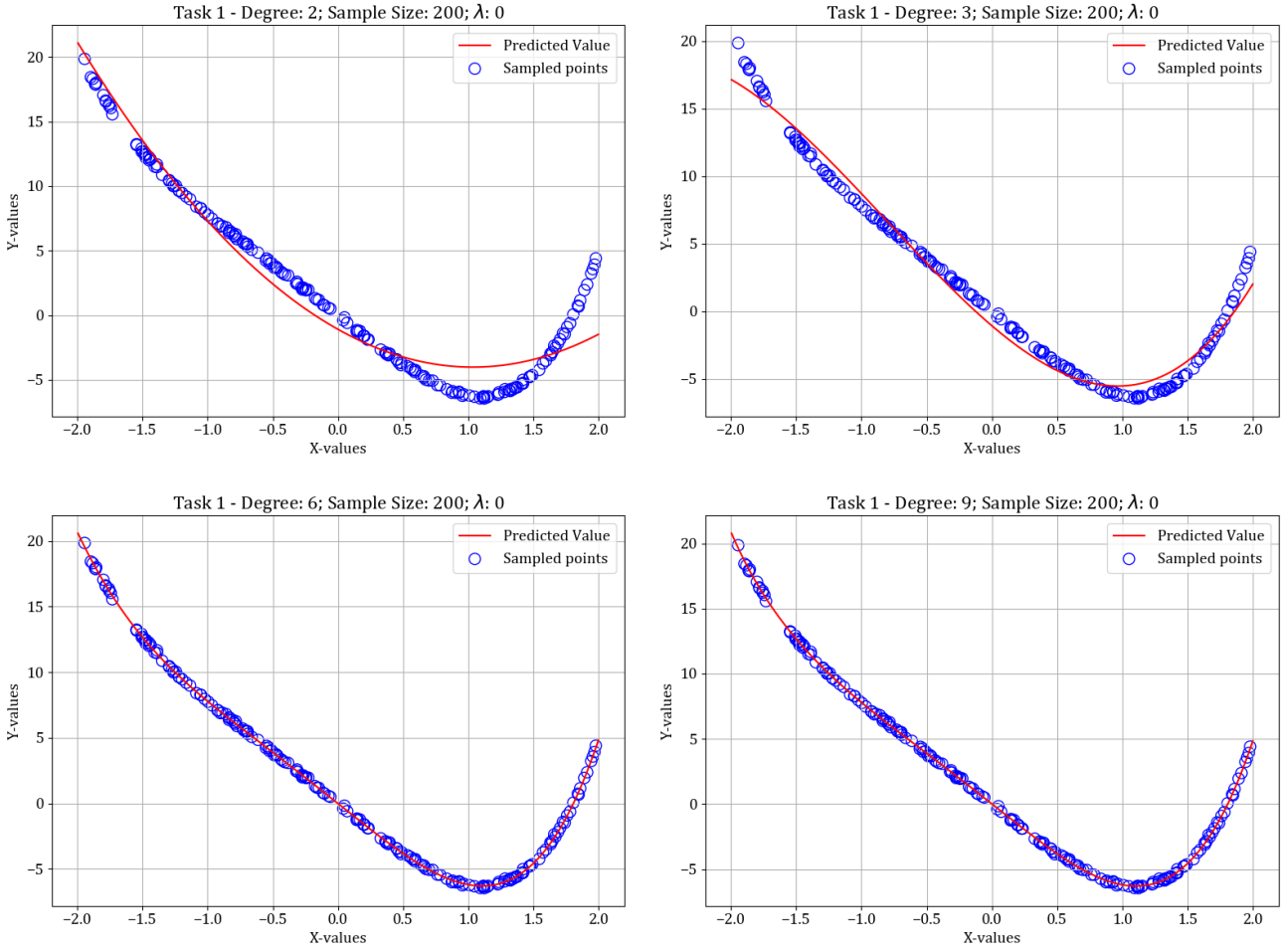


Figure 2: Task 1 - Polynomial fits, Sample size: 200

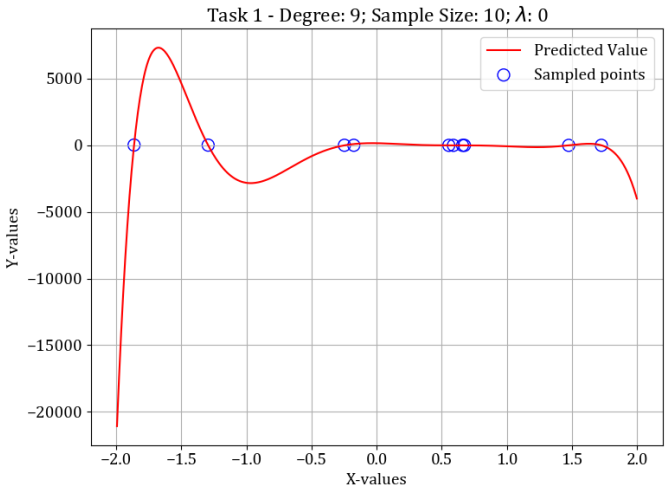
1.3.2.1 Inference

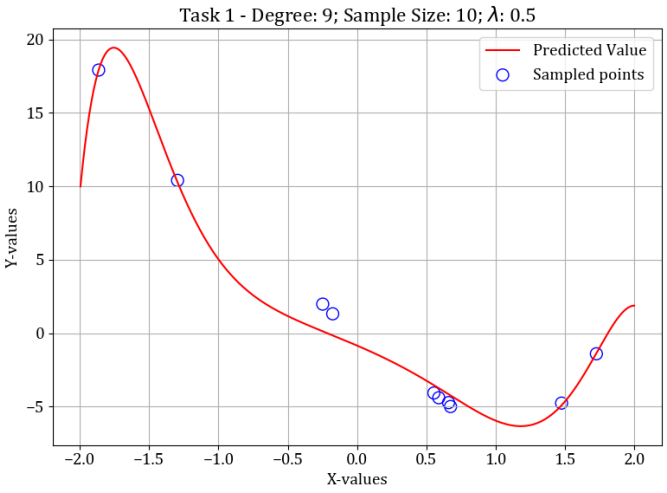
From the above plots, we can see that:

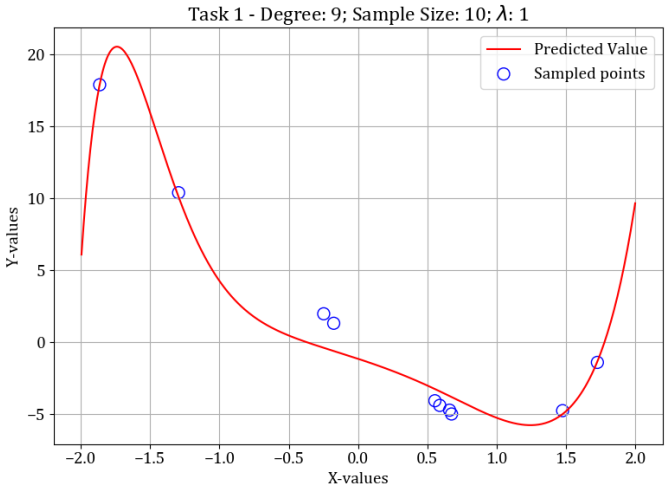
- Lower degree polynomial curves aren't able to model the dataset well (i.e.) the curve doesn't pass through all the data points.
- Higher degree polynomials are able to fit the dataset well. The curves pass through all the data points.
- We can see a clear difference between the degree 9 fit when the dataset size was 10 to that when the dataset size is 200. The increase in dataset size helped decrease the variance ans potential overfitting.

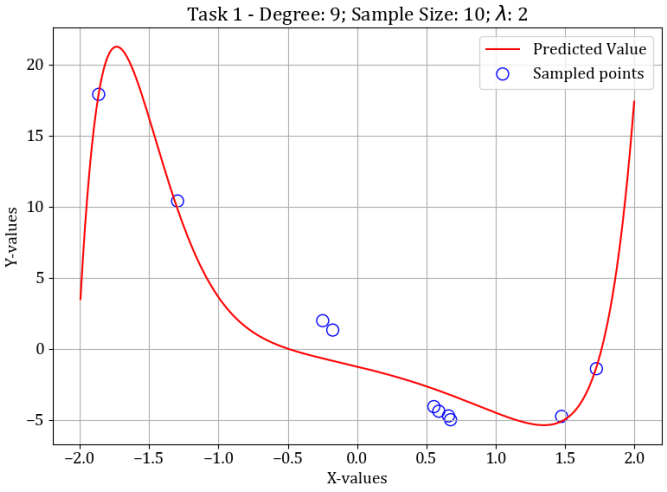
1.3.3 Effects of Regularization

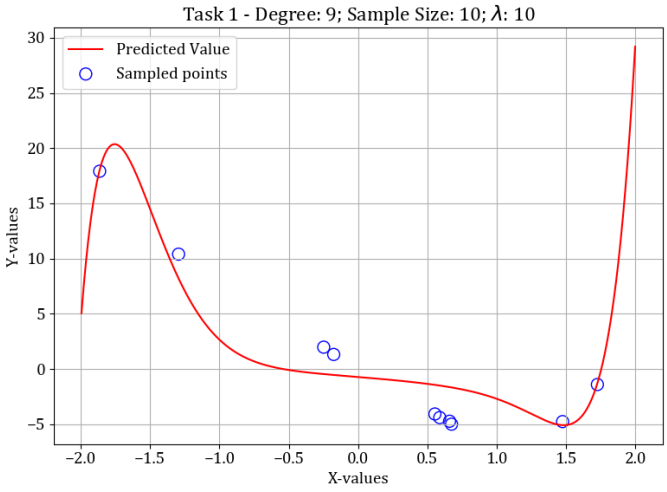
The polynomial models and the corresponding fits obtained for degree 9, sample size of 10, across different λ values are as follows:

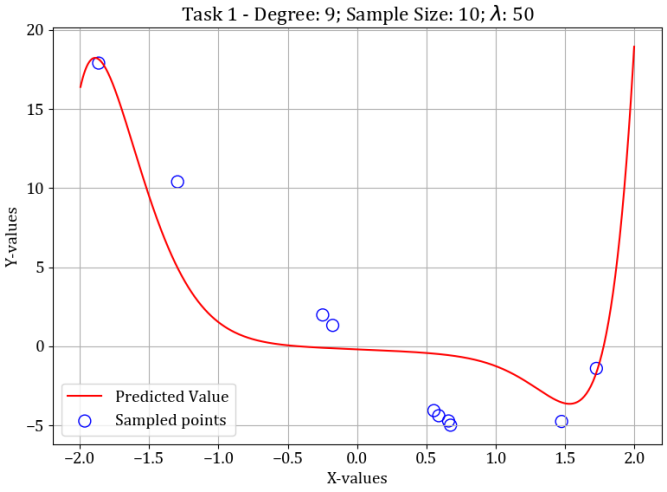












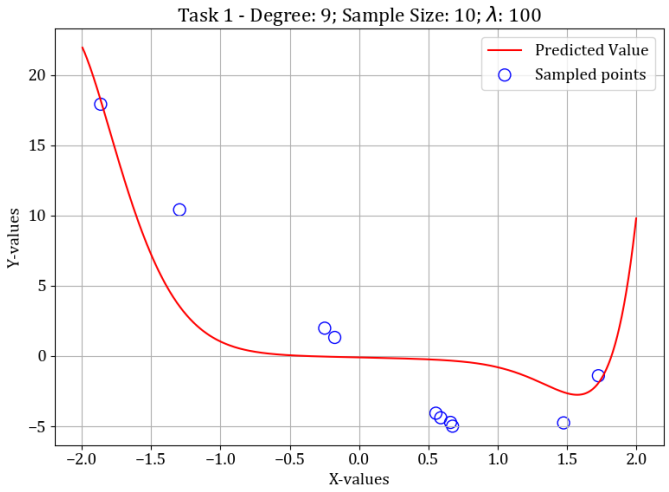


Figure 3: Task 1 - 9th Degree Polynomial fit, Sample size: 10

1.3.3.1 Inference

From the above plots, we can see that:

- Regularization was only applied to the degree 9 polynomial, with 10 data points as it had the same number of data points and parameters.
- We can see that, the curve starts becoming more flatter with increasing value of the regularization parameter λ.
- This could be because, the weights corresponding to higher degrees would become smaller.

1.4 Best Model

The best fit, $d : 6$ and $\lambda : 0$ is visualized as follows:

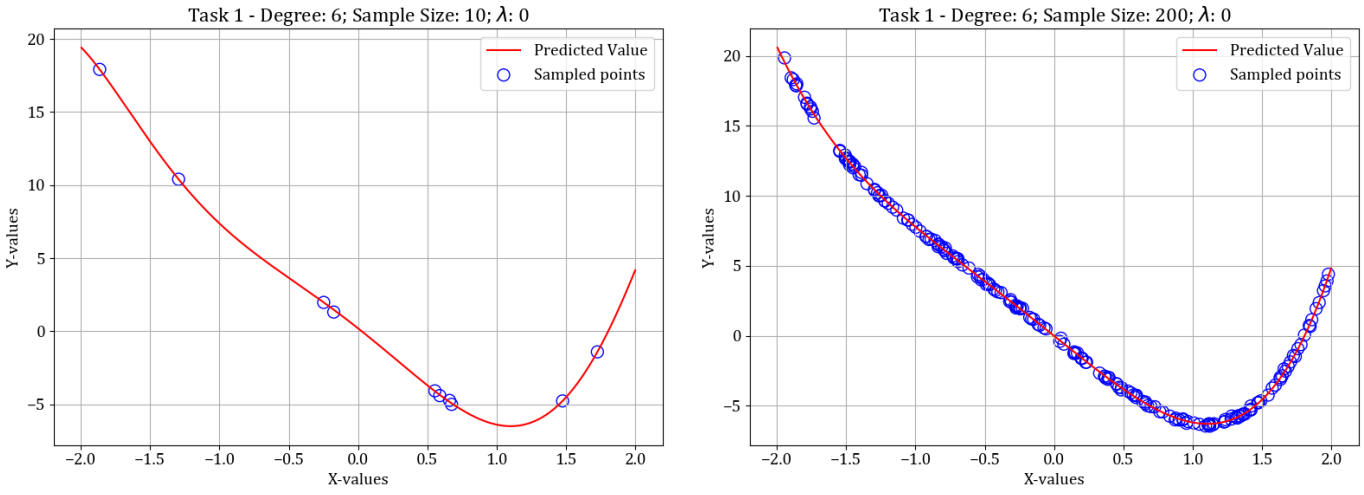


Figure 4: Task 1 - Best fit, Sample size: 10 (to the left) and Sample size: 200 (to the right)

The final training and testing error obtained is as follows:

- Training Error: 0.09974659089780814
- Testing Error: 0.09793071099285168

2 Task 2

The dataset allotted to our group for task 2 is `function1_2d.csv`, which has a 2 dimensional feature vector and 1 dimensional target output to be predicted. We assume that the target variable is of the form:

$$y = \sum_{i=0} \omega_i \phi_i(x1, x2) + \epsilon \tag{3}$$

Where ω_i are the parameters to be found through regression, $\phi_i(x1, x2)$ is a polynomial in x1 and x2 and ϵ is the normally distributed error.
A breakdown of the steps undertaken is:

- The function `create_phi` generates the design matrix $\phi(x1, x2)$ for the required degree of complexity. The number of attributes in the generated design matrix is given by:

  $$n = \frac{(d+D)!}{d!\,D!} \tag{4}$$

  where d is the dimension of the original feature vector (=2 for Task 2) and D is the degree of complexity of the model.

- The design matrix is passed to the function `regularized_pseudo_inv` , which generates the moore-penrose inverse of the given design matrix(X) and specified value of regularization parameter lambda(λ).

  $$(\lambda I + X^T X)^{-1} X^T \tag{5}$$

- The function `opt_regularized_param` is then used to obtain optimum values of $\vec{\omega}$

  $$\vec{\omega} = [(\lambda I + X^T X)^{-1} X^T].y \tag{6}$$

  Where y is the output as defined in the equation 3.

- The optimum parameter vector thus obtained can be used to predict the variable y for new inputs.

  $$y_{prediction} = X\vec{\omega} \tag{7}$$

The results obtained for various degrees of complexities are discussed below.

2.1 Degree of complexity = 2

With degree of complexity set to 2, the number of parameters in our model are:

$$\begin{aligned} n &= \frac{(d+D)!}{d!\,D!} \\ &= \frac{(4!)}{2!\,2!} \\ &= 6 \end{aligned} \tag{8}$$

Since the number of parameters to be estimated is very less compared to our sample sizes, we do not expect to see over fitting, and hence regularisation is not used.

2.1.1 Surface plots of Approximated function

Surface plots obtained for various train sizes are as follows:

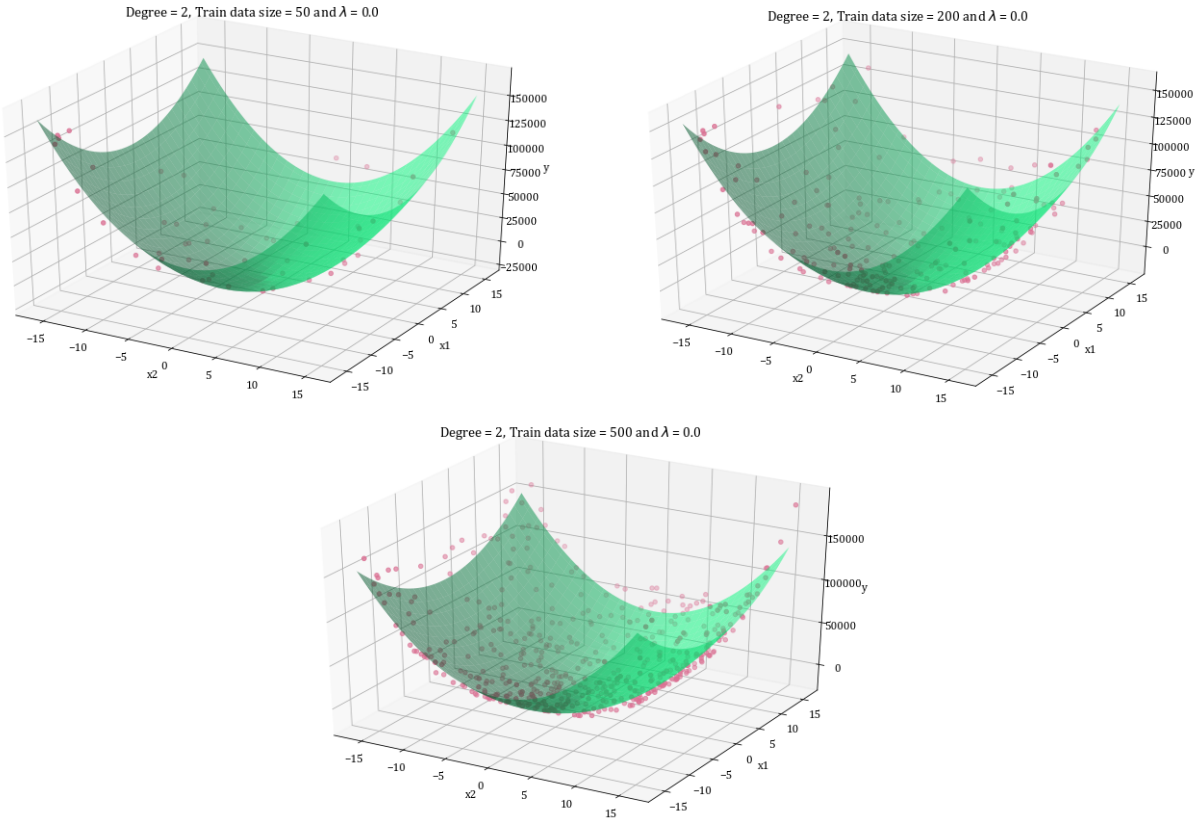


Figure 5: Surface Plot of the approximated function for different training sizes, Degree: 2

2.1.2 Erms over Train, Validation and Test data

The Erms over train, validation and test data is obtained to be:

Train size	λ	Erms Train	Erms Validation	Erms Test
50	0	9.34*10^3	1.06*10^4	1.14*10^4
200	0	1.06*10^4	1.14*10^4	1.15*10^4
500	0	1.13*10^4	1.12*10^4	1.08*10^4

Table 3: Erms for different train sizes for degree of complexity 2

2.1.3 Observation

- While the magnitude of E_{rms} is nearly same over train, validation and test data, it does not reduce on increasing the sample size.
- The surface plot of approximated function is simple and poorly fits both the training as well as test data.
- From the above two points, we conclude that we have an oversimplified model with a high bias. Increasing the complexity would be beneficial.

2.2 Degree of complexity = 3

The number of parameters to be estimated for this model are:

$$\begin{aligned} n &= \frac{(2+3)!}{2!\,3!} \\ &= 10 \end{aligned} \tag{9}$$

For this model too, the number of parameters to be estimated are very less compared to the train data sizes, and hence regularization is not required.

2.2.1 Surface plots of the approximated function

The surface plots of approximated function for various train data sizes is:

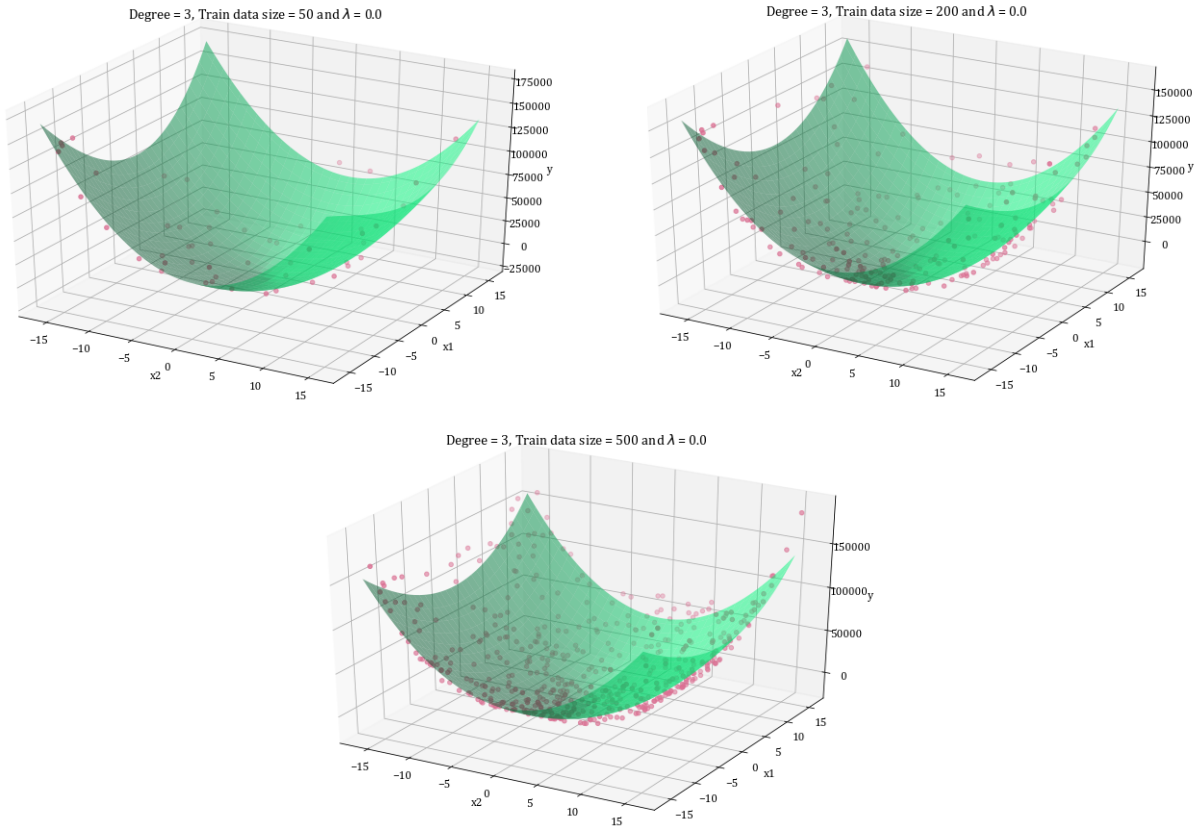


Figure 6: Surface plot of approximated function for different train sizes, Degree: 3

2.2.2 Erms over Train, Validation and Test data

The E_{rms} over Train, Validation and Test data is obtained to be:

Train size	λ	Erms Train	Erms Validation	Erms Test
50	0	8.40*10^3	1.19*10^4	1.23*10^4
200	0	1.03*10^4	1.14*10^4	1.15*10^4
500	0	1.11*10^4	1.11*10^4	1.11*10^4

Table 4: Erms for different train sizes for degree of complexity 3

2.2.3 Observation

- The E_{rms} values are nearly same as that for degree of complexity 2.
- Increasing the sample size does not affect the E_{rms} significantly.
- While E_{rms} Train is lower for sample size 50, it is due to inadequate number of data samples. E_{rms} Train, E_{rms} Validation and E_{rms} Test converge as the train data size increases to 500.
- From the above points we conclude that this model too is oversimplified and thus fails to perform well over Train, Validation as well as Test data. Our model thus has a high bias error similar to model of complexity 2.

2.3 Degree of complexity = 6

The number of parameters to be estimated are-

$$\begin{aligned} n &= \frac{(2+6)!}{2!\,6!} \\ &= 28 \end{aligned} \tag{10}$$

For this model too, the number of parameters to be estimated is far less compared to train data size of 200 and 500. However, for the train data size of 50, we expect a poor estimation of the parameters since the model will not have enough data points.

2.3.1 Surface plots of the approximated function

The surface plots of the approximated function for various train data sizes are:

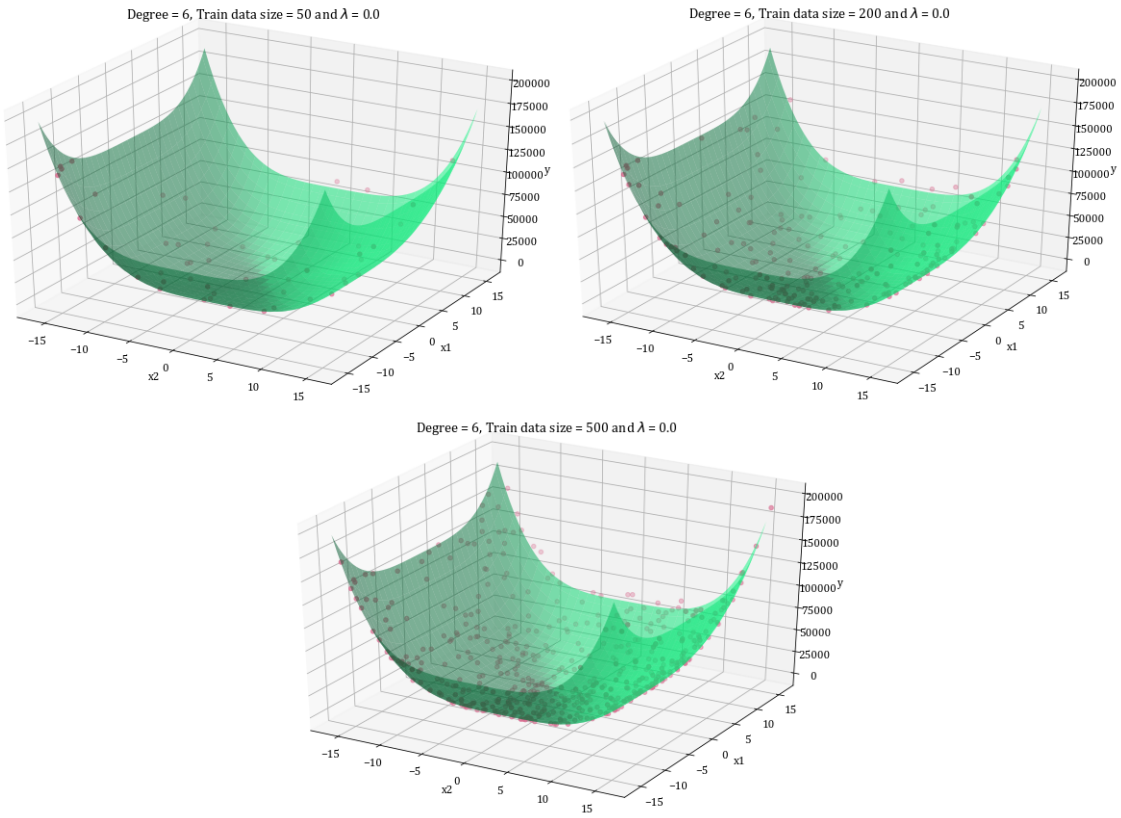


Figure 7: Surface plots of approximated function for different train size, Degree = 6

2.3.2 Erms over Train, Validation and Test data

The E_{rms} values obtained over Train, Validation and Test data are as follows:

Train size	λ	Erms Train	Erms Validation	Erms Test
50	0	7.78*10^{-8}	3.72*10^{-7}	6.17*10^{-7}
200	0	1.31*10^{-8}	1.39*10^{-8}	1.44*10^{-8}
500	0	3.47*10^{-8}	3.66*10^{-8}	3.39*10^{-8}

Table 5: Erms for different train sizes for degree of complexity 6

2.3.3 Observations

- The complexity of surface in Figure 7 has increased significantly compared to that in Figure 5 and Figure 6
- The E_{rms} values over all the data sets has decreased drastically as compared to the previous models.
- While the E_{rms} train is less compared to E_{rms} Validation and E_{rms} Test for train data size = 50, increasing the training data size alleviates this.
- On increasing the train data size to 200, E_{rms} over Train, Validation and Test data all converge to a lower value, signifying an optimum trade off between bias and variance error. Regularization is therefore not required.
- On further increasing the Train data size, the E_{rms} increases insignificantly.
- From the above points and cross-validation method, we conclude that the degree of complexity 6 and Train data size of 200 is the optimal model to describe our data, achieving an upper bound Root Mean Squared Error of $1.5 * 10^{-8}$ over Train, Validation as well as Test data.
- None of the models need to be regularized. On applying regularization, even for very small values of the hyperparameter λ, the E_{rms} errors increase.

2.4 Scatter plot of Model output vs Target output

Using the optimal model of degree 6 and train data size 200, model output vs target output was plotted for both Train and Test data, we find it to closely follow $y - x = 0$ line.

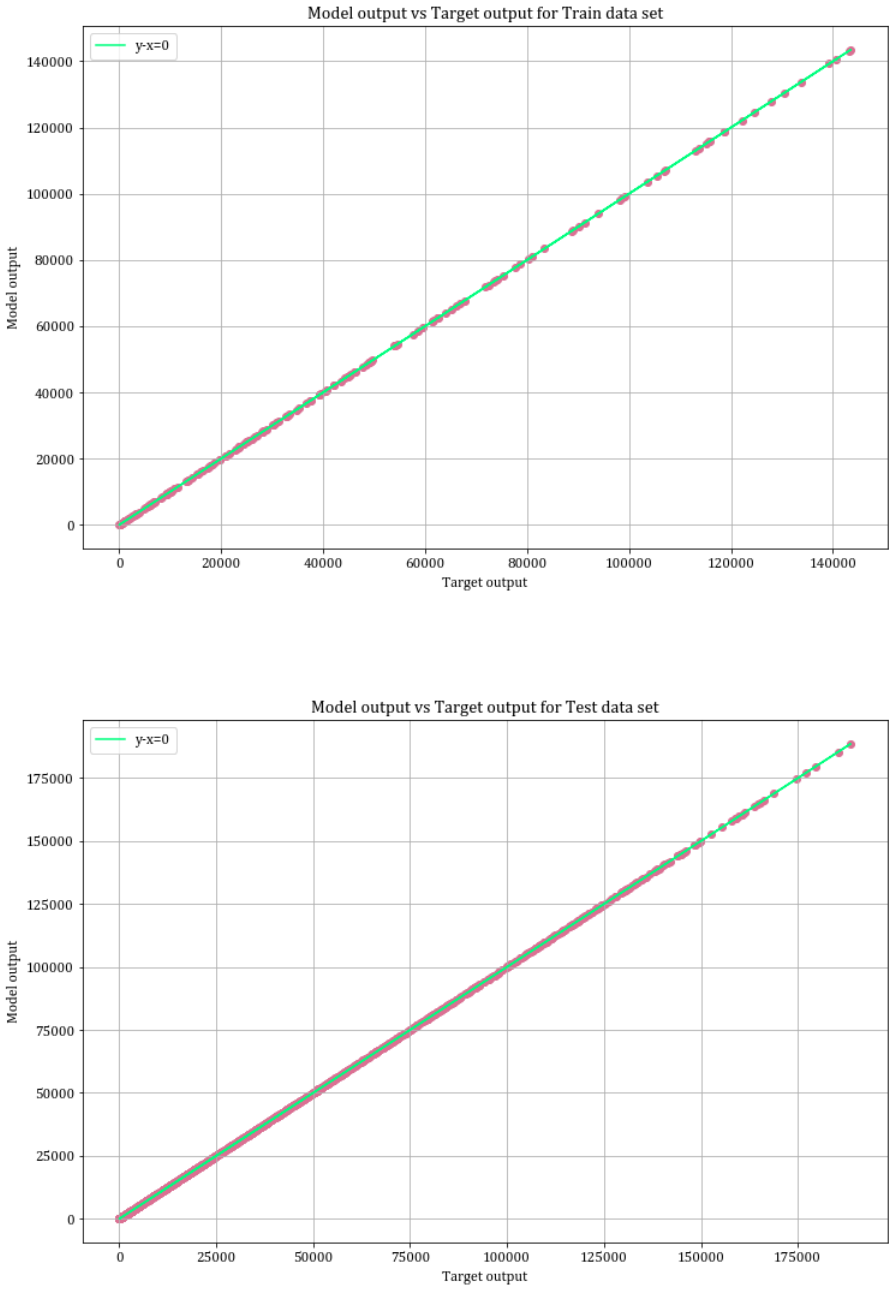


Figure 8: Model output vs Target output for train(left) and test dataset(right)

3 Task 3

Linear regression using Gaussian basis function is given as

$$y(\vec{x}, \vec{w}) = \sum_{i=0}^{D-1} \omega_i \phi_i(\vec{x}) \tag{11}$$

, where D is a hyperparameter. The basis function

$$\phi_i = \exp\left(\frac{-|\vec{x} - \vec{\mu}_i|^2}{\sigma^2}\right) \tag{12}$$

where $i = 1, 2...D-1$. The μ are the mean vectors for $D-1$ kernels made from the data set. The value of the mean vectors are found using the KMeans clustering algorithm. In this work, the `sklearn` KMeans function was used. The optimum number of clusters for the dataset 2 - "`function_12d.csv`" was found to be 10 clusters. For the dataset 3 - "`1_bias_clean.csv`", the optimum number of clusters are 9.

3.1 Dataset 2

3.2 Dataset 3

As the dataset was a real world dataset, the following preprocessing steps were carried out.

- `NaN` values: All datapoints that had `NaN` values in any of the columns were removed.

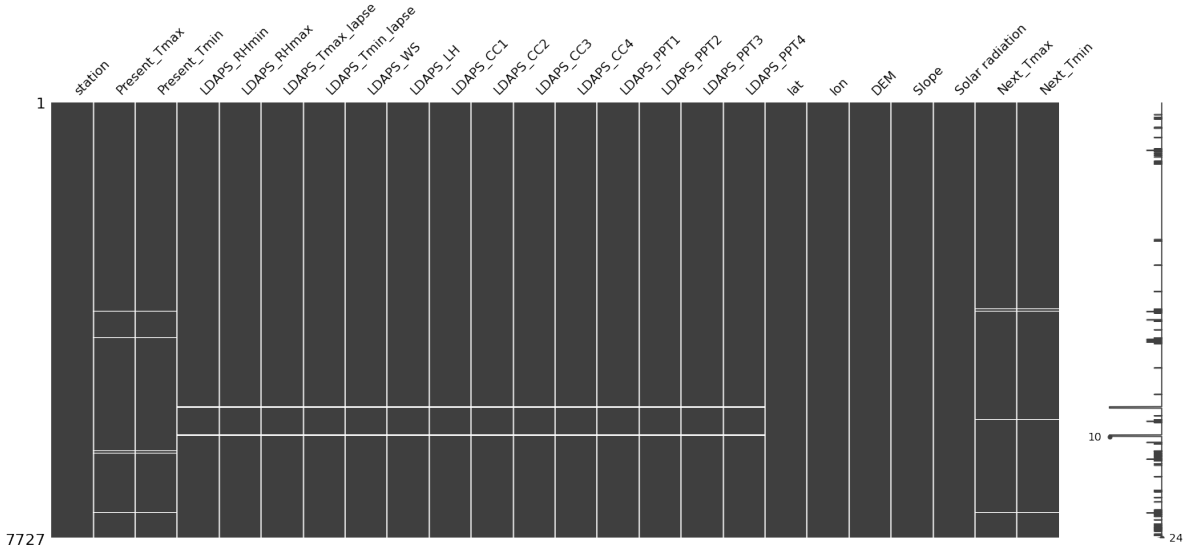


Figure 9: Visualization of the original dataset. White lines indicate `NaN`s.

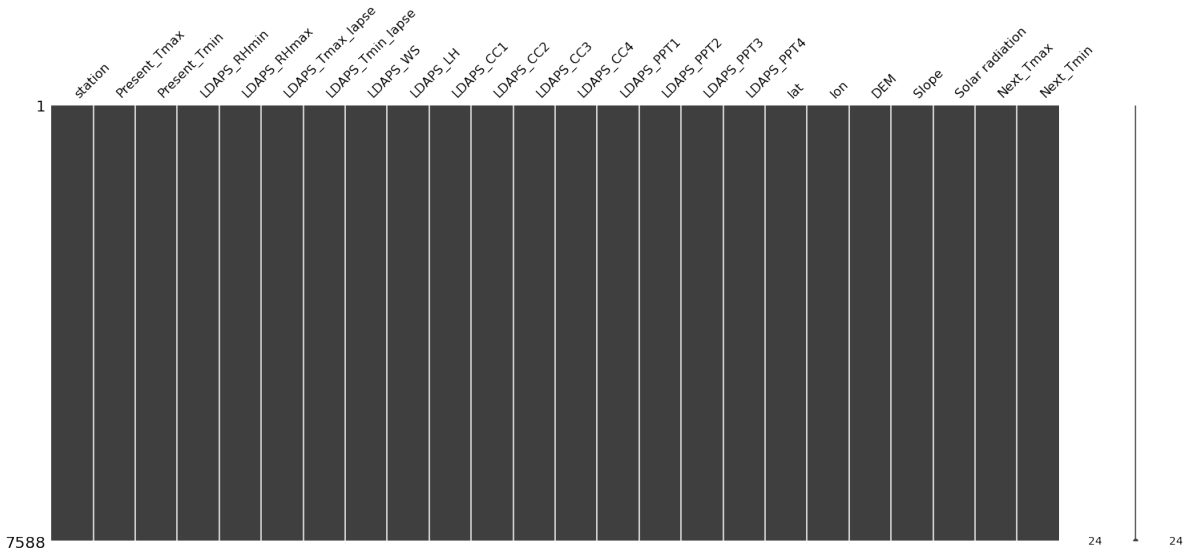


Figure 10: Visualization of the dataset after the removal of `NaNs`.

- Highly correlated factors were removed. A threshold of 0.75 was used to identify highly correlated features and they were sequentially removed.
- Features that resulted in a high Variance Inflation Factor (VIF) were also removed.
- The analysis involving correlated features and VIF was performed on the training data and was then extended to the validation and testing data.

3.2.1 No Regularization

The hyperparameter - number of clusters was sweeped and the value that resulted in the lowest validation SSE was choosen. The following cluster numbers were swept for: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 20, 25, 3$

3.2.1.1 Predicting: `Next_Tmin`

The SSE error on the training and validation dataset and the SSE distances of samples to their closest cluster center obtained across the number of clusters is as follows:

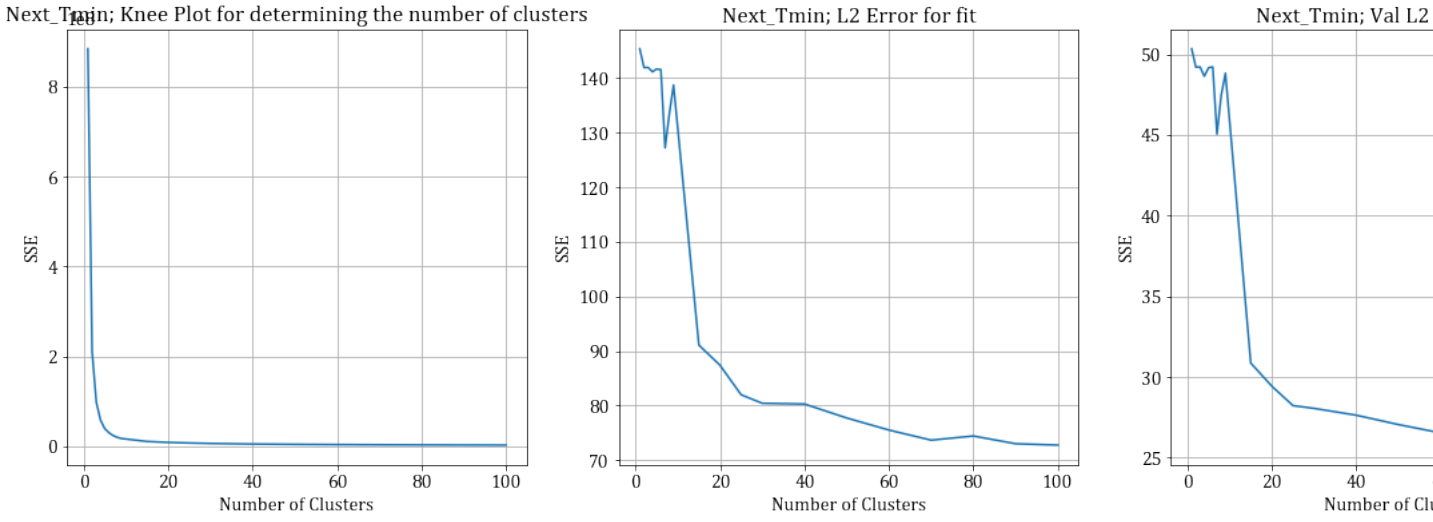


Figure 11: K-Means inertia, SSE on training and validation data from the left to right respectively.

The scatter plot of the target output and the model output is as follows:

Lambda	RMSE Train	RMSE CV	RMSE Test
0.01	3059.2231939706166	304419.6502059433	1309688.1097631603
0.1	2967.5181829927037	1260.488802049498	41195.93571016699
1.0	2990.3948869258456	1425.9232970077237	39596.78510421749
5.0	3013.546633117982	1503.9615829712025	39006.81341800579
10.0	3036.360553338793	1541.3504903884834	38684.11470986189

Table 6: Results obtained for Task 3

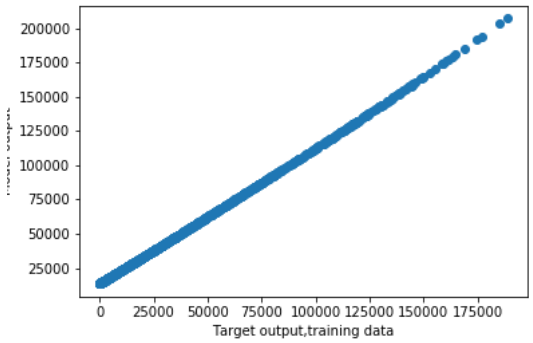


Figure 12: Scatter plot of the target values vs model prediction for Training set of Dataset 2, using linear regression with gaussian basis and no Regularization, $\lambda = 0.01$

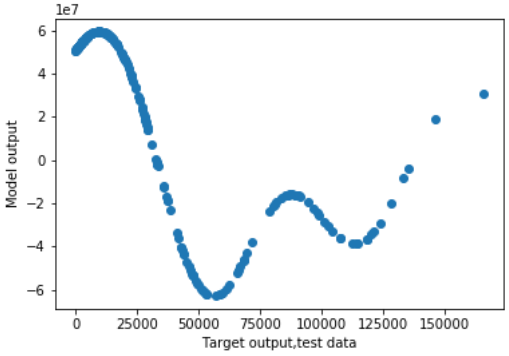


Figure 13: Scatter plot of the target values vs model prediction for Test set of Dataset 2, using linear regression with gaussian basis and no Regularization, $\lambda = 0.01$

3.3 Quadratic Regularization

Optimal parameters using quadratic regularization is given by $\vec{\omega^*} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\vec{t}$;

λ is the regularization parameter. The values $0.01, 0.1, 1.0, 5.0, 10.0$ were used to estimate the optimal parameters and the RMSE on the cross-validation set was calculated for each value. The best performing model was selected as the one having least RMSE on CV data

For dataset 2, the RMSE values for the Training, CV and Test data across λ values is:
Scatter plots of the model prediction using the regularization parameter value 0.01:

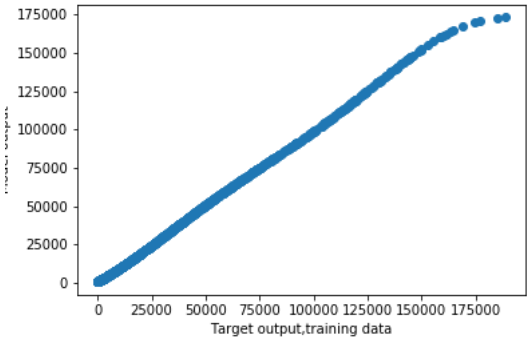


Figure 14: Scatter plot of the target values vs model prediction for Training set of Dataset 2, using linear regression with gaussian basis and quadratic Regularization, $\lambda = 0.01$

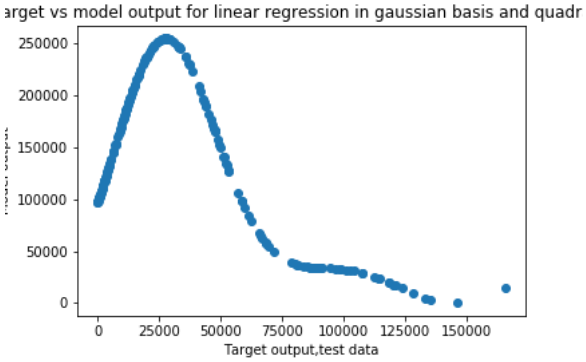


Figure 15: Scatter plot of the target values vs model prediction for Test set of Dataset 2, using linear regression with gaussian basis and quadratic Regularization, $\lambda = 0.01$

For dataset 3, the following RMSE table was obtained:

Lambda	RMSE Train	RMSE CV	RMSE Test
0.01	4929.01653053444	414848.3042013735	3312196.0992440097
0.1	4624.091536077471	1669.205290237633	36902.34473960538
1.0	4637.926167783597	2073.9375316355663	39568.04405922714
5.0	4652.374704231647	2257.2672544567195	40450.17809627787
10.0	4667.40003673528	2346.920490034092	40853.959722564214

Table 7: Results obtained for Task 3

The scatter plots of target vs model output for the optimum value of λ is, for ”NTmin” output variable

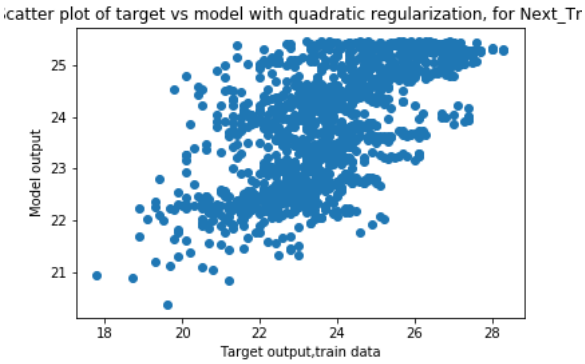


Figure 16: Scatter plot of the target values vs model prediction for Training set of Dataset 3, using linear regression with gaussian basis and quaadratic Regularization, $\lambda = 0.01$ for "NTmin" output variable

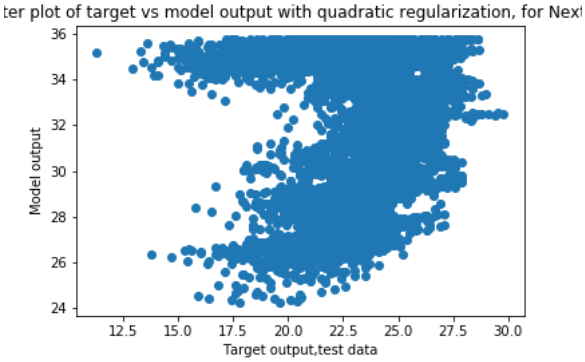


Figure 17: Scatter plot of the target values vs model prediction for Test set of Dataset 2, using linear regression with gaussian basis and quadratic Regularization, $\lambda = 0.01$, for "NTmin" output variable

For "NTmax":

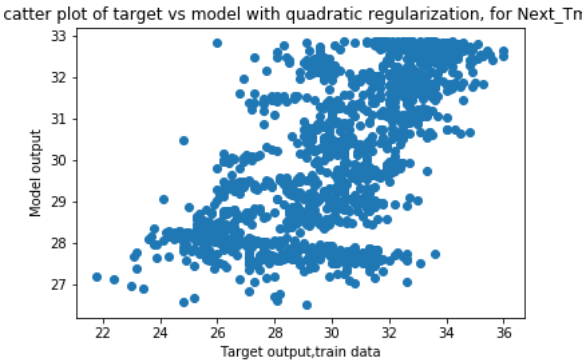


Figure 18: Scatter plot of the target values vs model prediction for Training set of Dataset 3, using linear regression with gaussian basis and quaadratic Regularization, $\lambda = 0.01$ for "NTmax" output variable

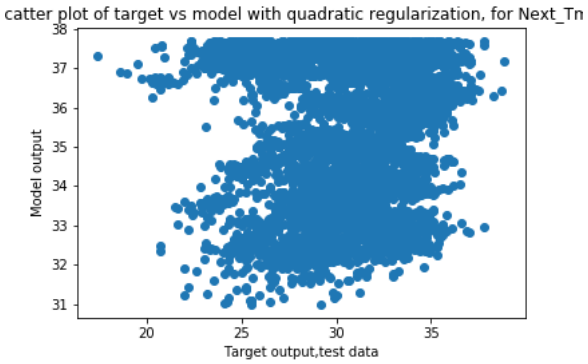


Figure 19: Scatter plot of the target values vs model prediction for Test set of Dataset 2, using linear regression with gaussian basis and quadratic Regularization, $\lambda = 0.01$, for "NTmax" output variable

3.4 Tikhonov Regularization

The Tikhonov regularization term is given by $\vec{\omega^*} = (\Phi * T\Phi + \lambda\tilde{\Phi})^{-1}\Phi^T\vec{t}$. The $\tilde{\Phi}$ term is defined as

$$\tilde{\Phi} = [\tilde{\phi}]_{i,j=1}^{K} \tag{13}$$

where K is the number of clusters and λ is the regularization parameter. The values 0.01,0.1, 1.0,5.0,10.0 were used to estimate the optimal parameters and the RMSE on the cross-validation set was calculated for each value. The best perorming model was selected as the one having least RMSE on CV data

Applying Tikhonov regularization to the bivariate dataset, the optimal value of λ was estimated to be 0.01. The table for the RMSE values for the Training, CV and Test values corresponding to each λ value is

Lambda	RMSE Train	RMSE CV	RMSE test
0.01	78112040.25241715	80416219813.42682	43898825037.538
0.1	78629878.58895023	276304216179.1798	176596027875.55765
1.0	79471104.52781227	357583741029.9827	104372422450.83612
5.0	77593937.82583737	272444591755.6569	174802698267.85687
10.0	77693846.61754198	242869557997.72043	158196054705.92648

Table 8: Results obtained for Task 3

Scatter plots of the model prediction using the regularization parameter value 0.01:

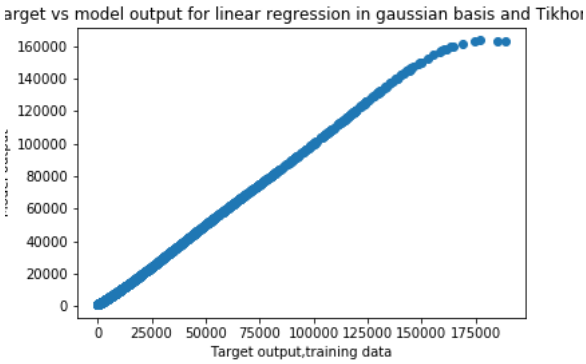


Figure 20: Scatter plot of the target values vs model prediction for Training set of Dataset 2, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.01$

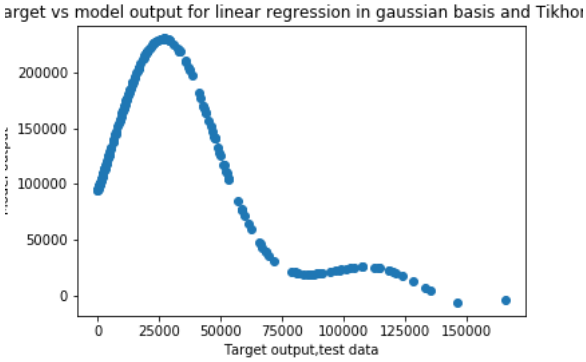


Figure 21: Scatter plot of the target values vs model prediction for Test set of Dataset 2, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.01$

For Dataset 3, the table for the RMSE values for the Training, CV and Test values corresponding to each λ value corresonding to target variable "NTmin" is

Lambda	RMSE Train	RMSE CV	RMSE test
0.01	4929.01653053444	414848.3042013735	3312196.0992440097
0.1	4624.091536077471	1669.205290237633	36902.34473960538
1.0	4637.926167783597	2073.9375316355663	39568.04405922714
5.0	4652.374704231647	2257.2672544567195	40450.17809627787
10.0	4667.40003673528	2346.920490034092	40853.959722564214

Table 9: Results obtained for Task 3

the optimal value of λ was estimated to be 0.1 for the target output "NTmin". The scatter plots obtained are Figure 22 and Figure 23

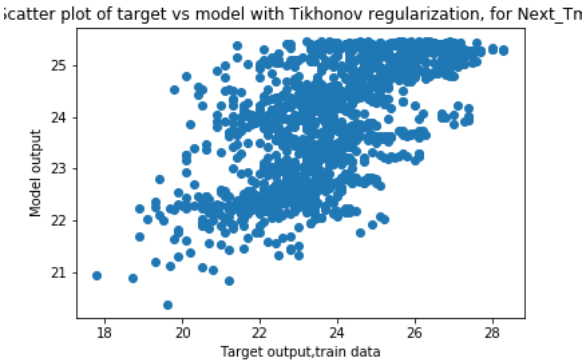


Figure 22: Scatter plot of the target values vs model prediction for Training set of Dataset 3, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.1$

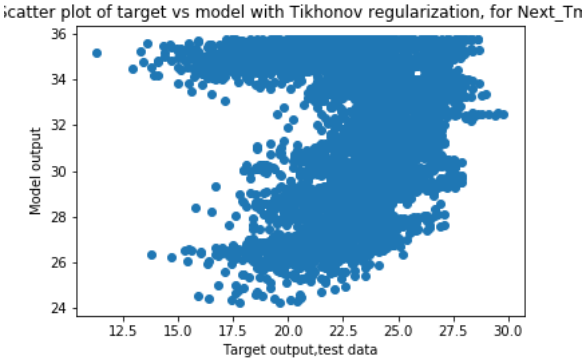


Figure 23: Scatter plot of the target values vs model prediction for Test set of Dataset 3, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.01$

For the target output "NTmax" the following table of RMSE values for the training, test and CV data was obtained:

Lambda	RMSE Train	RMSE CV	RMSE test
0.01	4929.01653053444	414848.3042013735	3312196.0992440097
0.1	4624.091536077471	1669.205290237633	36902.34473960538
1.0	4637.926167783597	2073.9375316355663	39568.04405922714
5.0	4652.374704231647	2257.2672544567195	40450.17809627787
10.0	4667.40003673528	2346.920490034092	40853.959722564214

Table 10: Results obtained for Task 3

plots were obtained: Figure 24 and Figure 25.

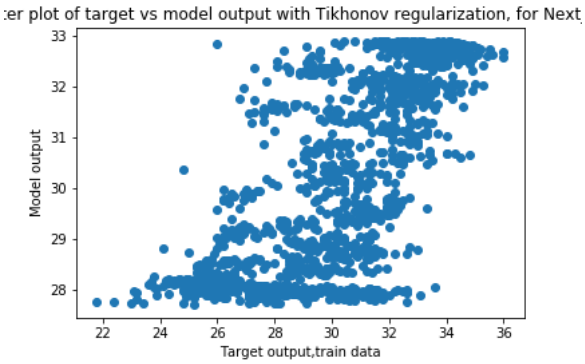


Figure 24: Scatter plot of the target values vs model prediction for Training set of Dataset 3, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.1$

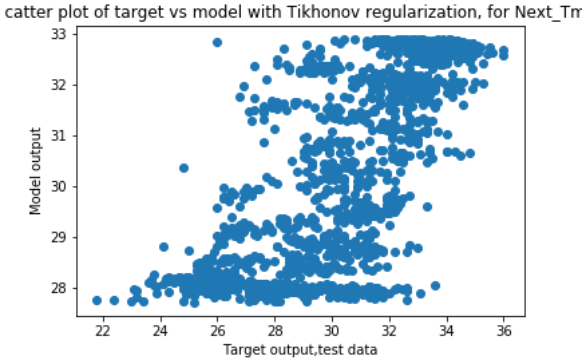


Figure 25: Scatter plot of the target values vs model prediction for Test set of Dataset 3, using linear regression with gaussian basis and Tikhonov Regularization, $\lambda = 0.1$