ASSIGNMENT 1

CS5691 Pattern Recognition and Machine Learning

CS5691 Assignment 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:

d	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

d	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

Regularization was only applied in case of degree 9.

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:

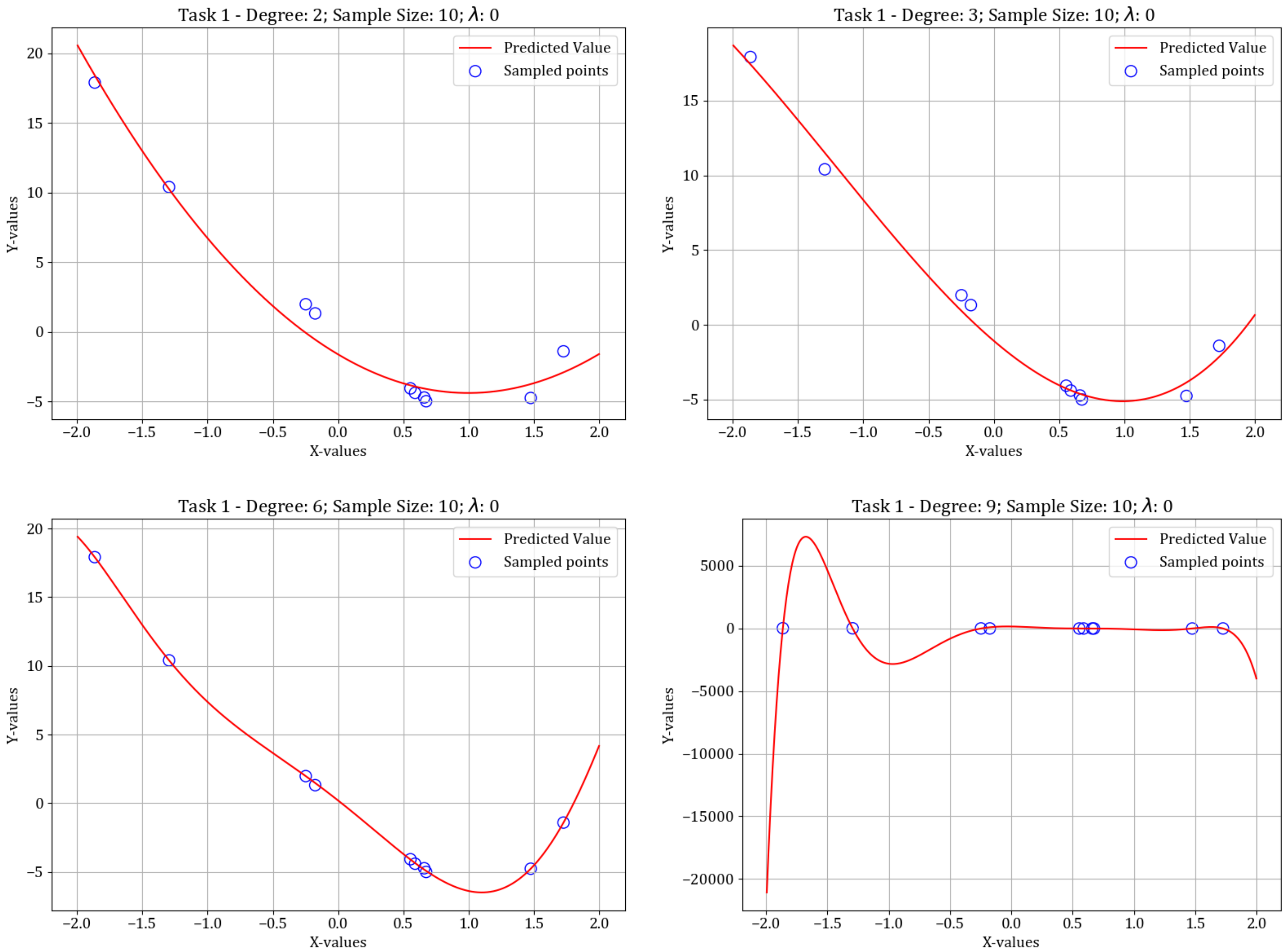


Figure 1: Task 1 - Polynomial fits, Sample size: 10

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 has a large 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:

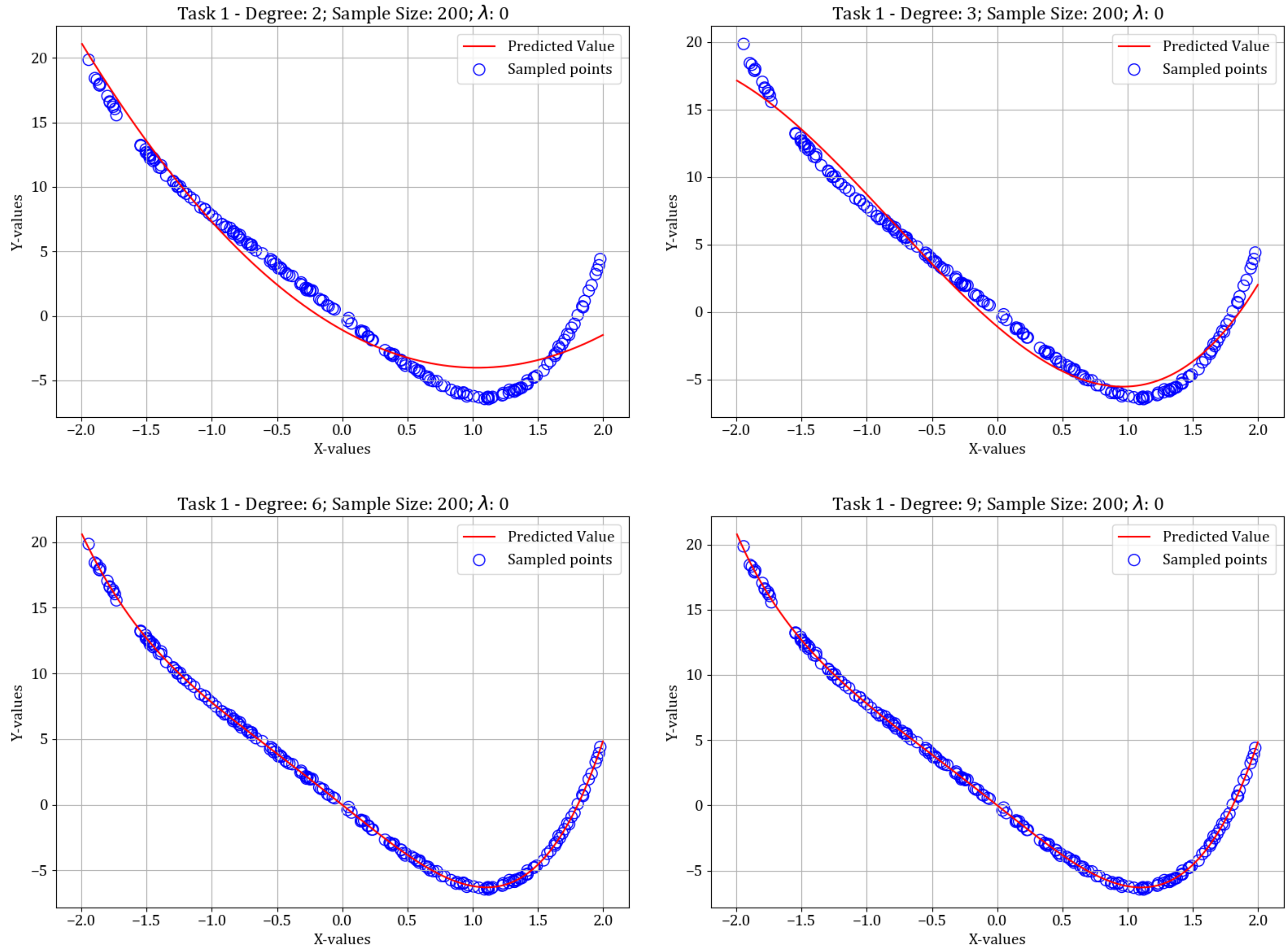


Figure 2: Task 1 - Polynomial fits, Sample size: 200

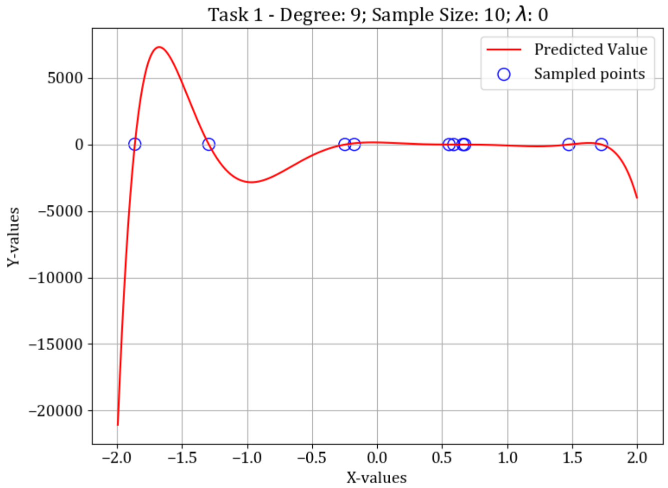
1.3.2.1 Inference

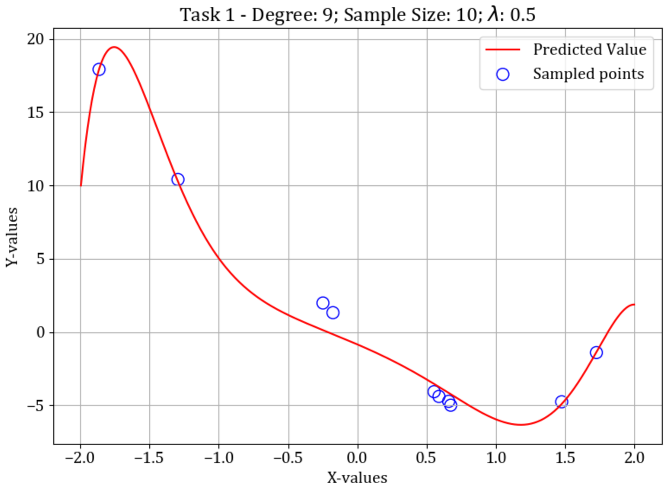
From the above plots, we can see that:

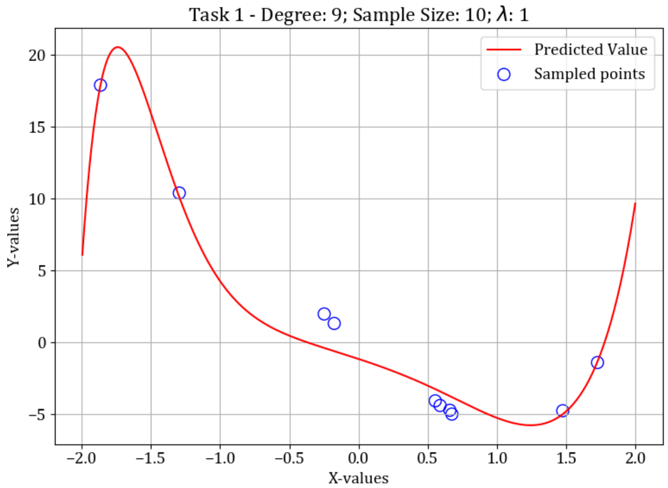
- Lower degree polynomial curves aren't able to model the dataset well (i.e.) the curve doesn't pass through all the data points.
- Higher degree polynomials are able to fit the dataset well. The curves pass through all the data points.
- We can see a clear difference between the degree 9 fit when the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

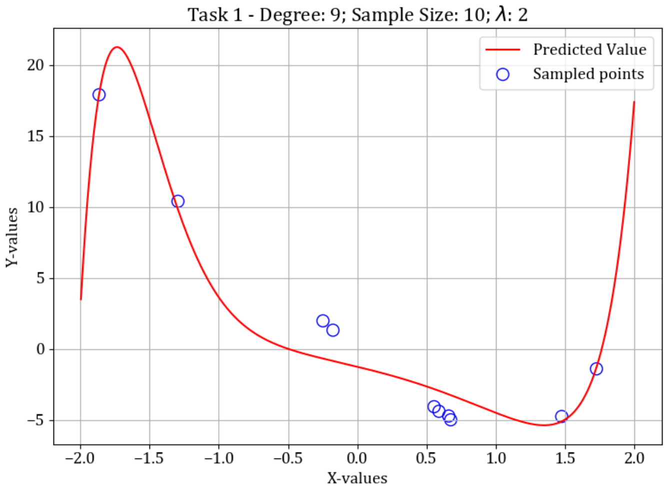
1.3.3 Effects of Regularization

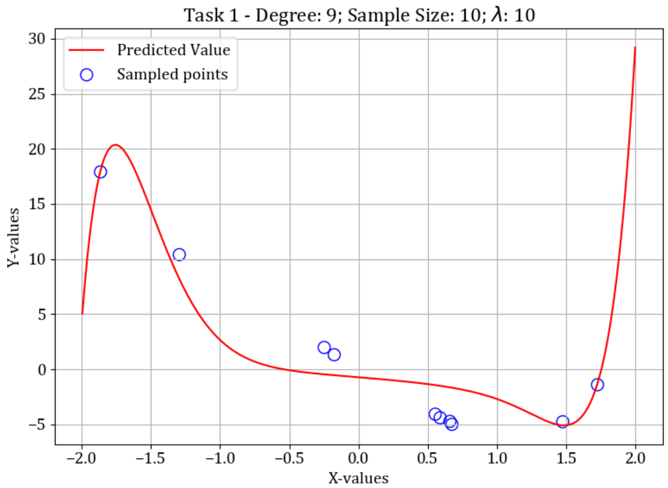
The polynomial models and the corresponding fits obtained for degree 9, sample size of 10, across different λ values are as follows:

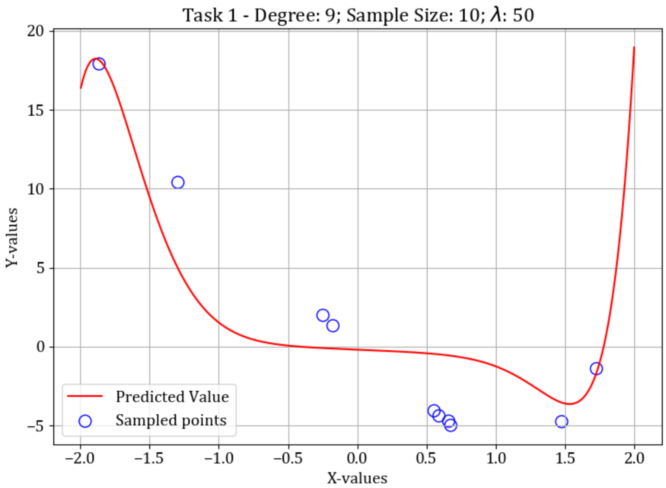

Task 1 - Degree: 9; Sample Size: 10; λ: 0
Predicted Value
Sampled points
Y-values
X-values


Task 1 - Degree: 9; Sample Size: 10; λ: 0.5
Predicted Value
Sampled points
Y-values
X-values


Task 1 - Degree: 9; Sample Size: 10; λ: 1
Predicted Value
Sampled points
Y-values
X-values


Task 1 - Degree: 9; Sample Size: 10; λ: 2
Predicted Value
Sampled points
Y-values
X-values


Task 1 - Degree: 9; Sample Size: 10; λ: 10
Predicted Value
Sampled points
Y-values
X-values


Task 1 - Degree: 9; Sample Size: 10; λ: 50
Predicted Value
Sampled points
Y-values
X-values

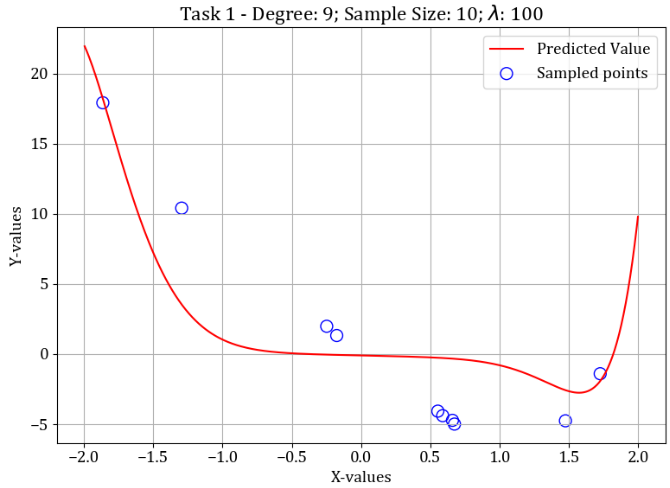


Figure 3: Task 1 - 9^{th} Degree Polynomial fit, Sample size: 10

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 became smaller.

1.4 Best Model

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

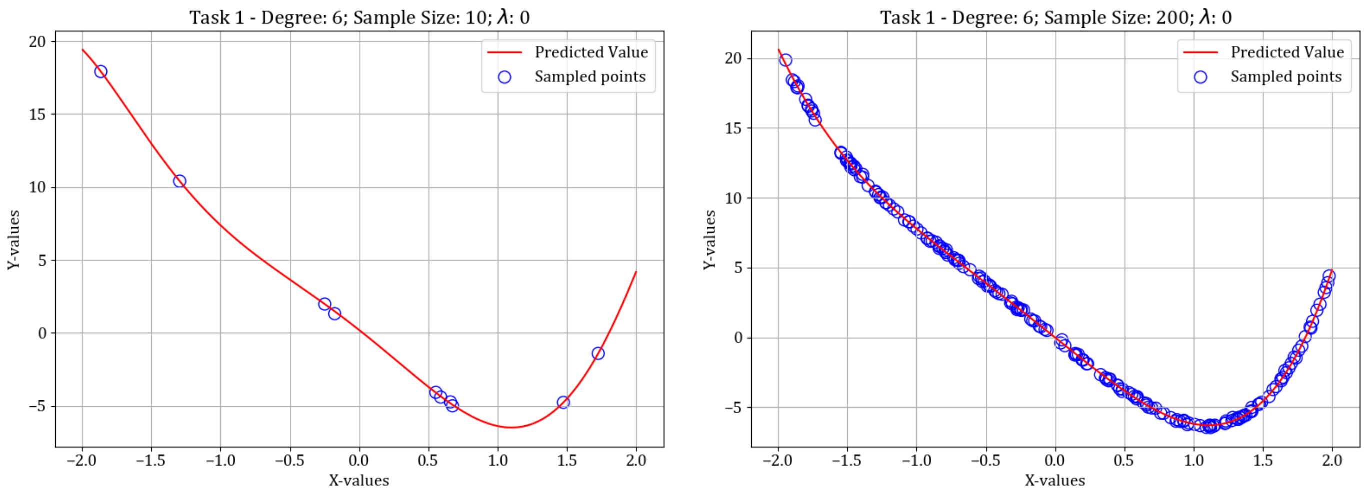


Figure 4: Task 1 - Best fit, Sample size: 10 (to the left) and Sample size: 200 (to the right)

The training and testing error obtained from the best model is as follows:

- Training Error: 0.09974659089780814
- Testing Error: 0.09793071099285168

2 Task 2

2.1 Mathematical Formulation

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:

$$y = \sum_{i=0} \omega_i \phi_i(x_1, x_2) + \epsilon \tag{3}$$

Where ω_i are the parameters to be found through regression, $\phi_i(x1, x2)$ is a polynomial in x_1 and x_2 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:

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

  where d is the dimension of the original feature vector (=2 for Task 2) and M is the degree of complexity of the model.
- To avoid overfitting, as a general rule `N` $> 10 * D$
- 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.

2.2 Degree of complexity: 2

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

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

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 regularization is not used.

2.2.1 Surface plots of Approximated function

Surface plots obtained for various train sizes are as follows:

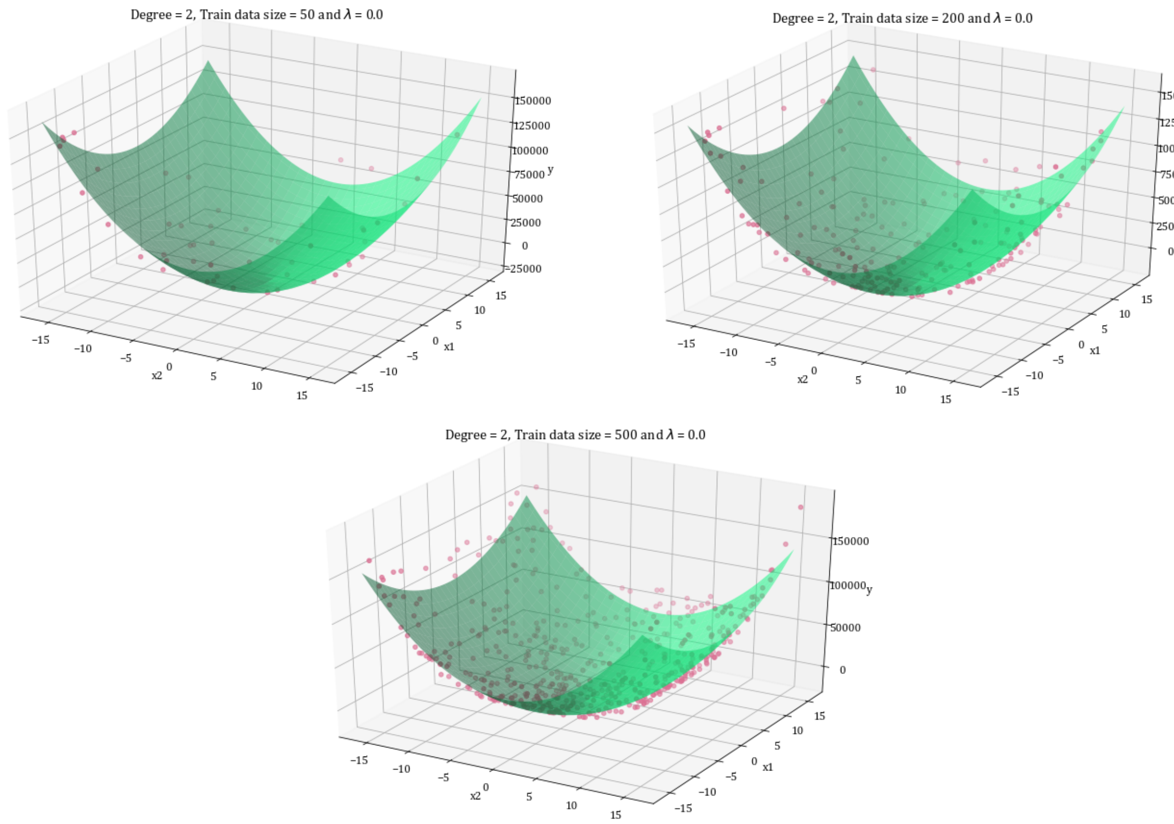


Figure 5: Surface Plot of the approximated function for different training sizes, Degree: 2

2.2.2 Erms over Train, Validation and Test data

The E_{rms} over train, validation and test data is obtained to be:

Train size	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **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: E_{rms} for different train sizes for degree of complexity 2

2.2.3 Inference

- 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.

2.3 Degree of complexity: 3

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

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

Since for train data size $50, 50 < 10 * 10$, we apply regularization. As reported in the E_{rms} table, the errors increases on applying regularization. Regularization need not be applied for train data size 200 and 500.

2.3.1 Surface plots of the approximated function

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

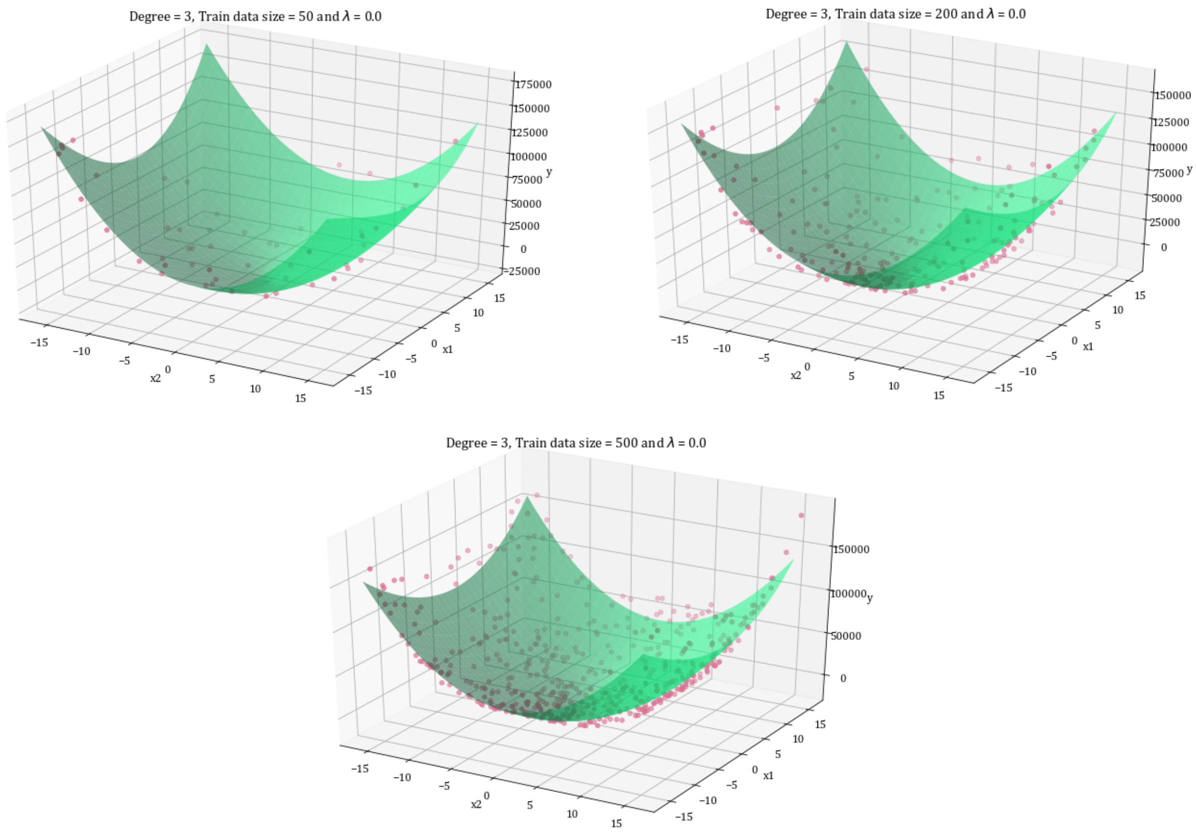


Figure 6: Surface plot of approximated function for different train sizes, Degree: 3

2.3.2 Erms over Train, Validation and Test data

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

Train size	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
50	0	8.40*10^3	1.19*10^4	1.23*10^4
50	1	8.43*10^3	1.19*10^4	1.22*10^4
50	10	9.24*10^3	1.07*10^3	1.31*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: E_{rms} for different train sizes for degree of complexity 3

2.3.3 Inference

- 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.

2.4 Degree of complexity: 6

The number of parameters to be estimated are-

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

For train data size 50, $N < 28 * 10$, hence regularization is applied to the model. However, regularization is not needed for training data sizes of 200 and 500.

2.4.1 Surface plots of the approximated function

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

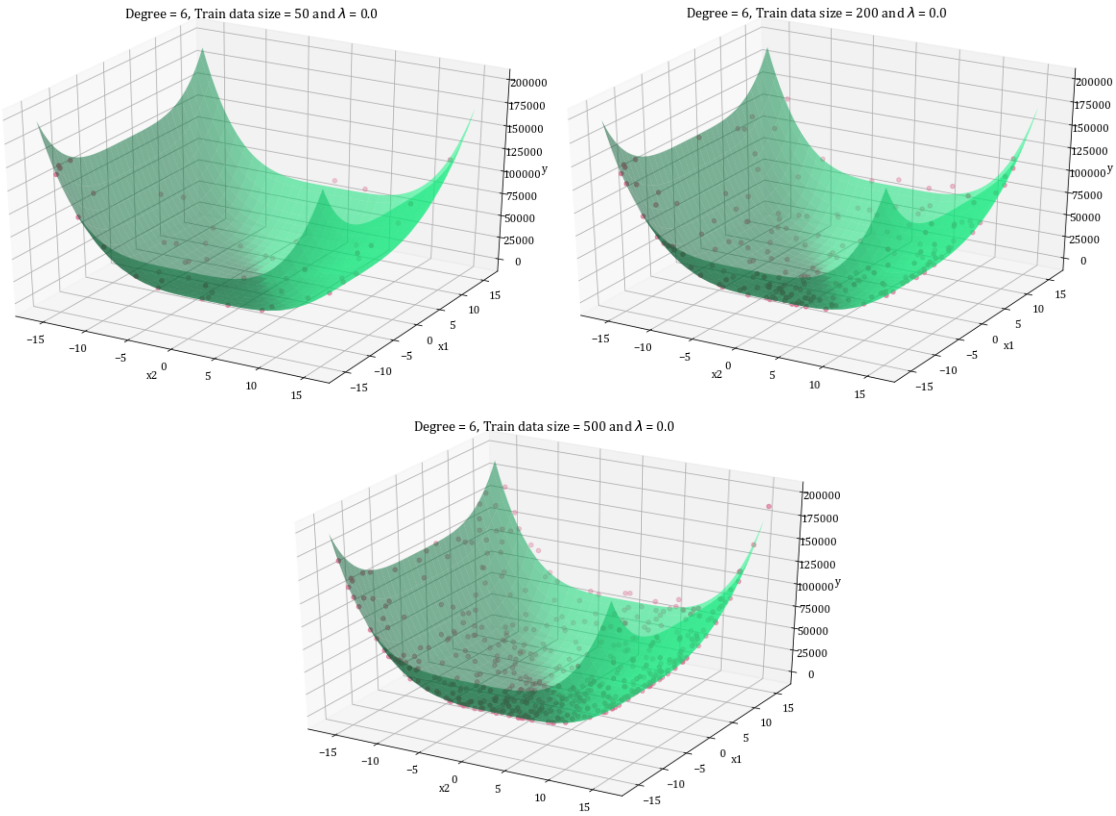


Figure 7: Surface plots of approximated function for different train size, Degree = 6

2.4.2 Erms over Train, Validation and Test data

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

Train size	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
50	0	7.78*10^{-8}	3.72*10^{-7}	6.17*10^{-7}
50	1	1.02*10^{-4}	1.25*10^{-3}	2.27*10^{-3}
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: E_{rms} for different train sizes for degree of complexity 6

2.4.3 Inference

- 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.

2.5 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.

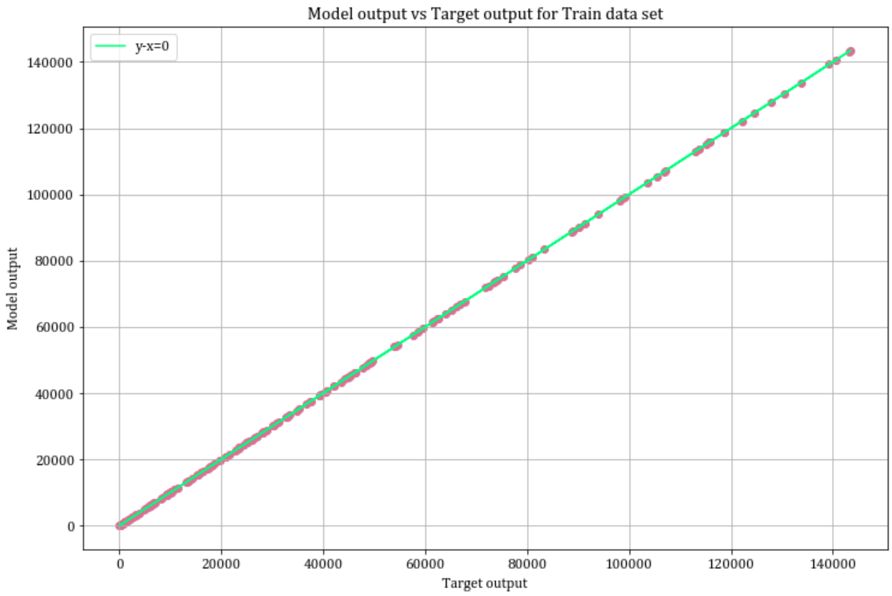


Figure 8: Model output against Target output for train dataset.

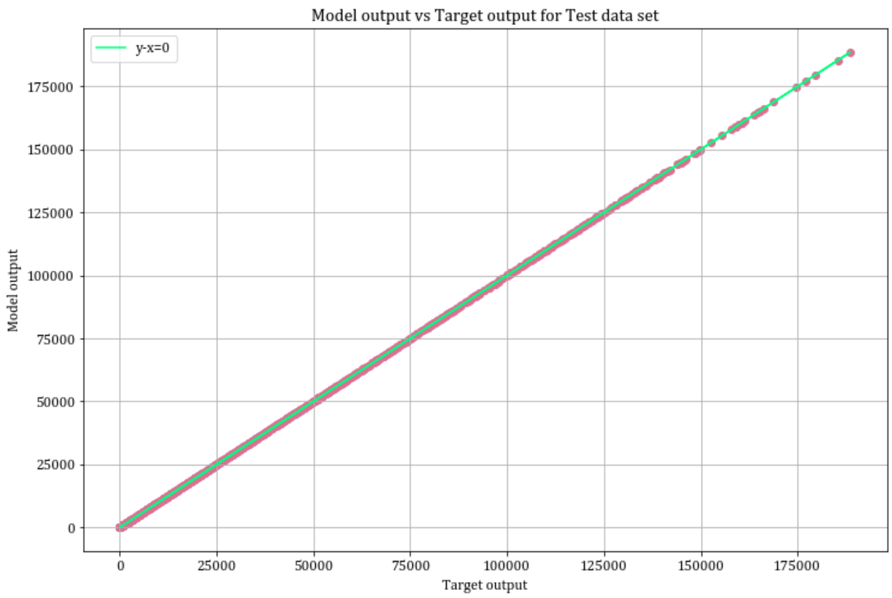


Figure 9: Model output against Target output for test dataset.

3 Task 3

3.1 Mathematical Formulation

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 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

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

and

$$\tilde{\phi}_{ij} = \exp\left(\frac{-|\vec{\mu}_i - \vec{\mu}_j|^2}{\sigma^2}\right) \tag{14}$$

where K is the number of clusters and λ is the regularization parameter. Running gridsearch on regularization parameter and number of clusters, the result is as given in the table

# Clusters	λ	SSE Train	SSE Validation	SSE Test
17	0.0	17367.7173058023	5476.451324628718	8586.275447090009
20	0.0	15739.95913019115	5640.71448410716	8203.349879534771
18	0.0	15871.388683543684	5806.4028863253025	8081.62361107947
16	0.0	22856.225772195507	8434.480223923712	11587.12003974188
15	0.0	41370.115688753016	14926.838846493623	19181.692764006595
21	0.0	51273.65291388557	16936.021162911442	27351.885426699668
22	0.0	55105.615267110115	19472.74104304696	28643.56695448404
24	0.0	76238.9270460689	27773.233458390918	39458.247076422114
19	0.0	95977.8642857218	32648.791441696205	50540.80341682174
14	0.0	119299.92496255055	46992.97792871992	60255.93503691771

Table 6: Top 10 SSE on the Training, Validation and Testing dataset, across different number of clusters, using tikhonov regularization for dataset 2.

3.2 Dataset 2

Using Gaussian-basis functions for the regression, three cases of regularization were carried out as follows:

3.2.1 No Regularization

With no regularization, the only hyperparameter is the number of clusters. Building models for number of clusters ranging from 1 to 26, the Sum of Squared errors for the train, cross-validation and test data are in Table 7.

Number of Clusters	Training Error	CV error	Test error
1	24162.18	9515.08	13461.11
2	23366.64	9149.58	12873.64
3	23356.33	9203.65	12956.92
4	23459.03	9245.98	13009.79
5	16892.92	6189.1	8918.64
6	14216.4	5412.48	8203.34
7	14112.21	5420.39	8280.16
8	13748.4	5278.88	8001.02
9	13754.34	5272.51	7966.53
10	13680.91	5247.02	7893.35
11	5908.12	2365.23	3138.67
12	5370.2	2048.94	2796.02
13	5592.17	2150.63	2922.27
14	4770.38	1879.97	2408.49
15	1655.91	597.64	770.59
16	944.08	352.7	480.17
17	694.6	218.4	343.12
18	582.56	186.93	308.31
19	786.11	284.72	406.82
20	866.17	281.13	468.87
21	13168.89	4435.47	6919.79
22	2204.22	778.91	1145.74
23	7496.9	2620.73	3891.07
24	15934.71	5227.49	8481.87
25	42105.16	15497.88	21925.28

Table 7: SSE error on dataset 2 using no regularization

The variation of cross-validation error with the number of clusters is as in Figure 10. We can infer from the table and the figure, the optimal number of clusters is **20**.

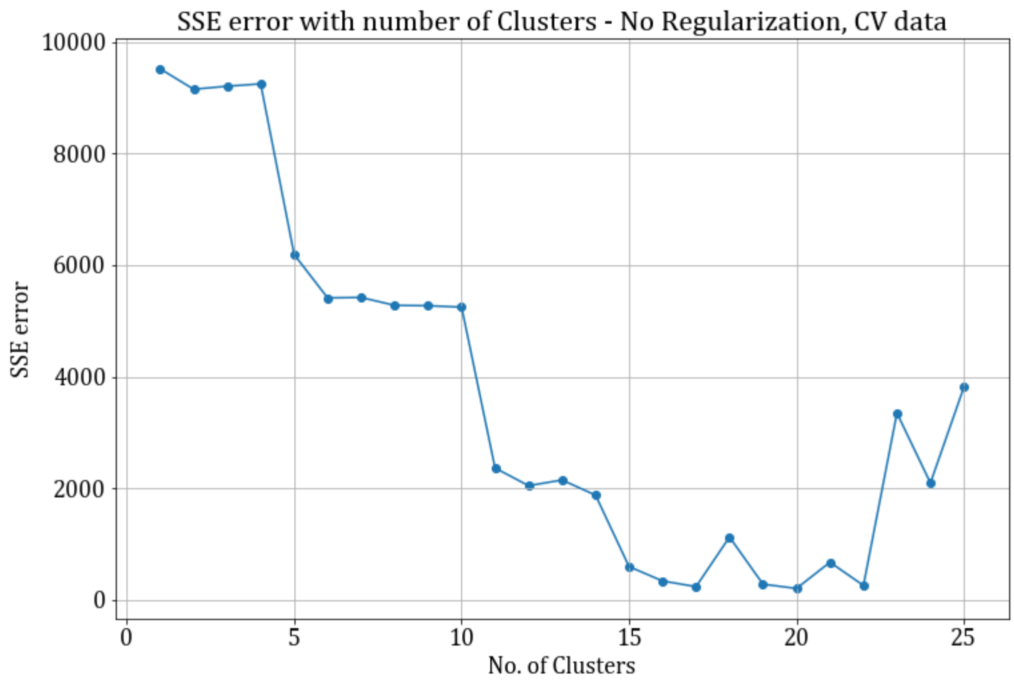


Figure 10: Plot of CV error with number of clusters, on dataset 2 and no regularization

Using the model with 20 clusters on the training and test data, the function in Figure 11 was built.

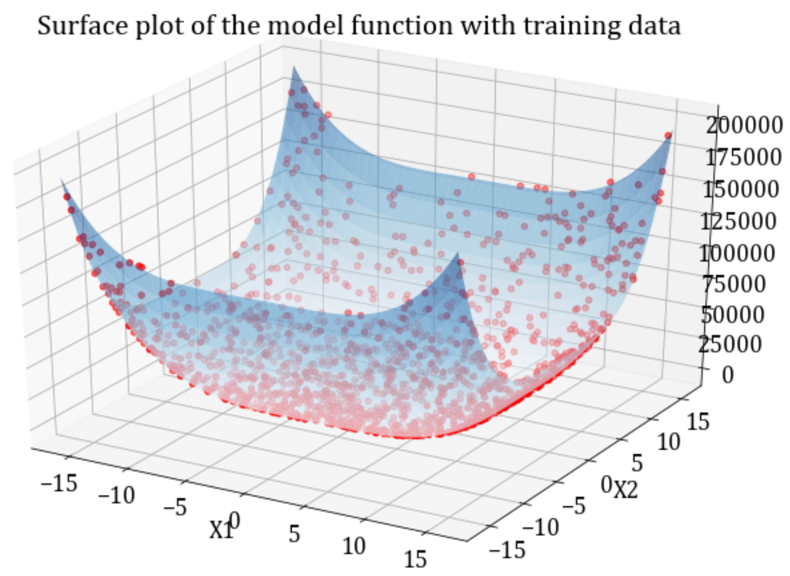
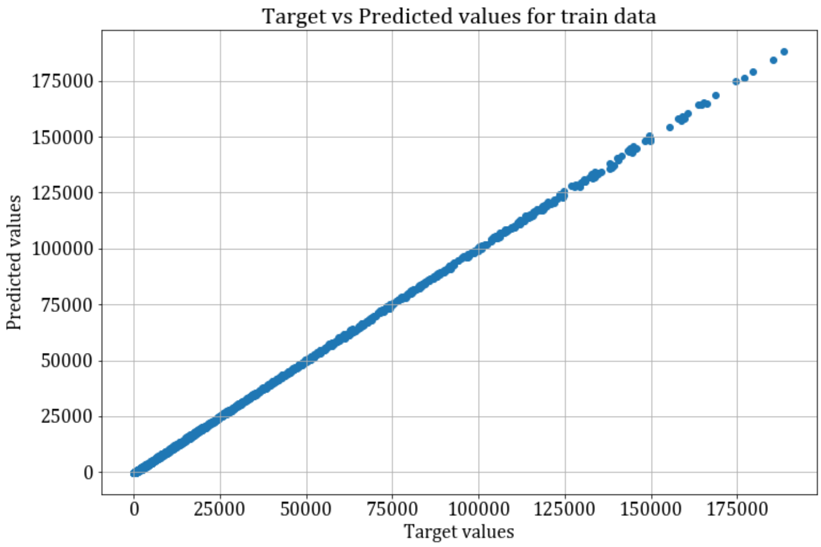


Figure 11: Best model of Gaussian basis with no regularization for dataset 2, with training data superimposed.

The performance on training and test data are shown in figure



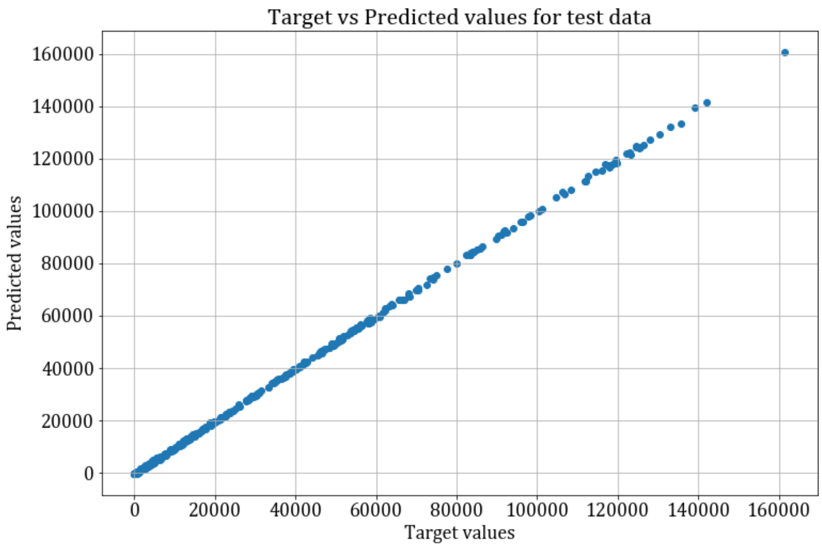


Figure 12: Predictions on train and test data using gaussian basis functions with no regularization on dataset 2

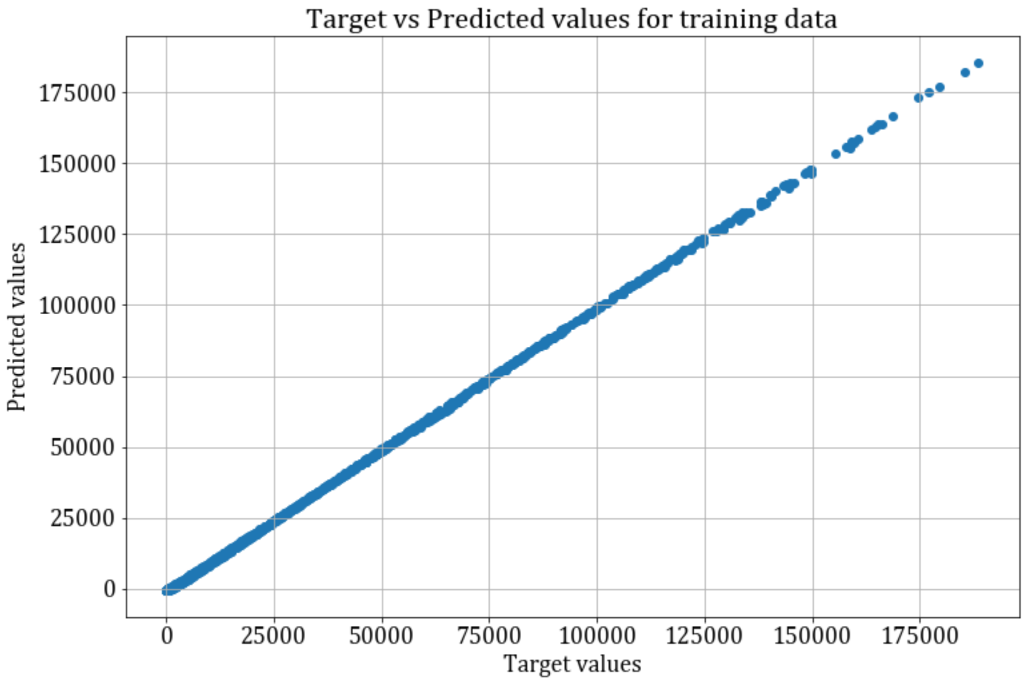
3.2.2 Quadratic regularization

Performing Quadratic regularization as described in section 2, a gridsearch was performed for the best regularization parameter and number of clusters, as given in table

# Clusters	λ	SSE Train	SSE Validation	SSE Test
18	0.0	14563.95	4673.21	7707.75
20	0.0	14377.72	5204.0	7345.81
17	0.0	19580.33	6732.68	9912.01
16	0.0	22856.23	8434.48	11587.12
19	0.0	29072.8	10207.35	15265.15
15	0.0	41578.12	14999.85	19067.9
22	0.0	74304.87	23487.19	39912.58
14	0.0	119373.62	47199.11	60138.86
12	0.0	134254.98	51223.42	69900.59
24	0.0	138137.32	52442.5	69461.67

Table 8: Top 10 SSE on the Training, Validation and Testing dataset, across different number of clusters, using L2 regularization for dataset 2.

The best model on validation data does not require any regularization. The optimum number of clusters is similar to the no regularization case of 20. Applying this model on the training and test data, we get the Figure 13



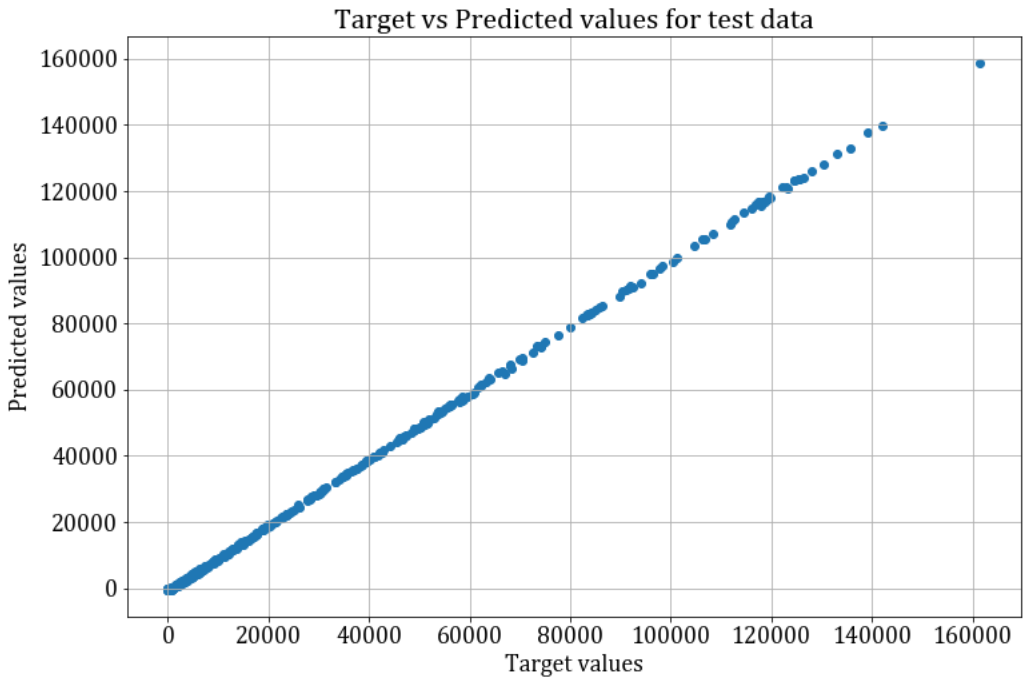
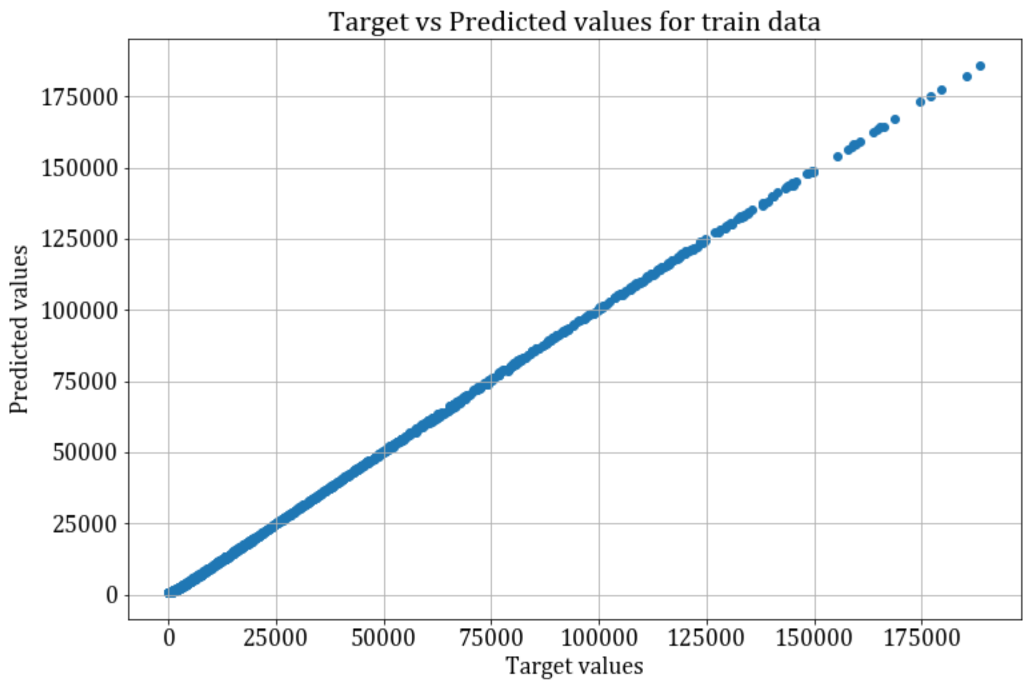


Figure 13: Gaussian basis model with 18 clusters on dataset 2

3.2.3 Tikhonov regularization

We can see in the tables that no regularization is required, and the optimal number of clusters is 17 (similar to the previous two cases). Applying this model on training and test data, we get the Figure 14



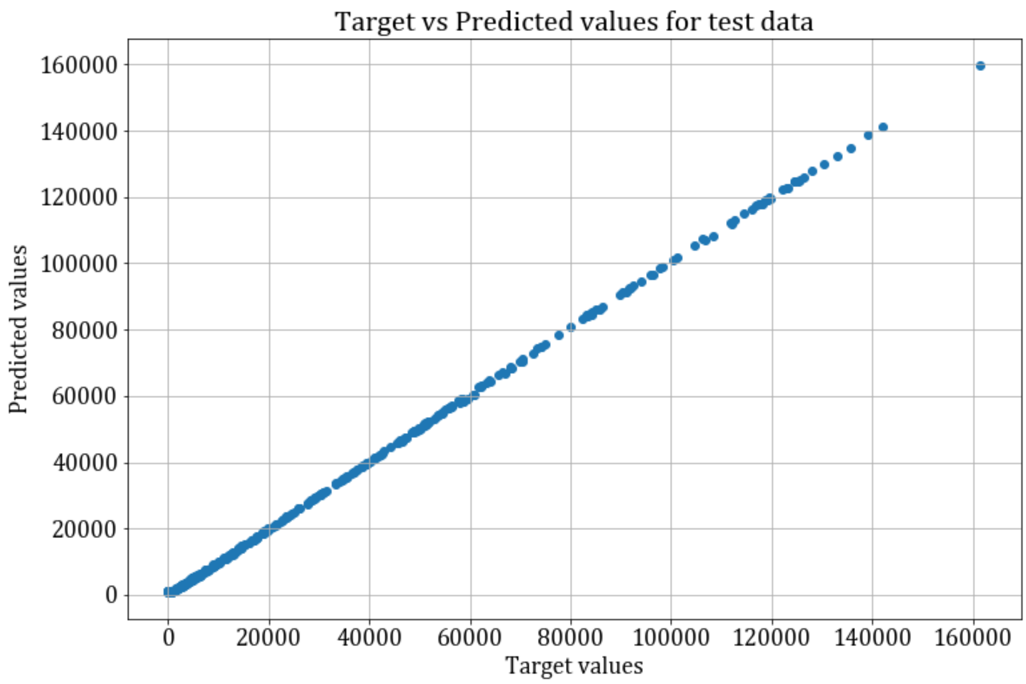


Figure 14: Gaussian basis model with 17 clusters on dataset 2

3.2.4 Inference

Comparing the different regularization methods, we can see that there is no overfitting on the data. Applying any form regularization on the full data worsens the errors.

The lowest E_{rms} values obtained are as follows:

- Training E_{rms}: 1.4786466027685028
- Testing E_{rms}: 1.4982621391957776

3.3 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.

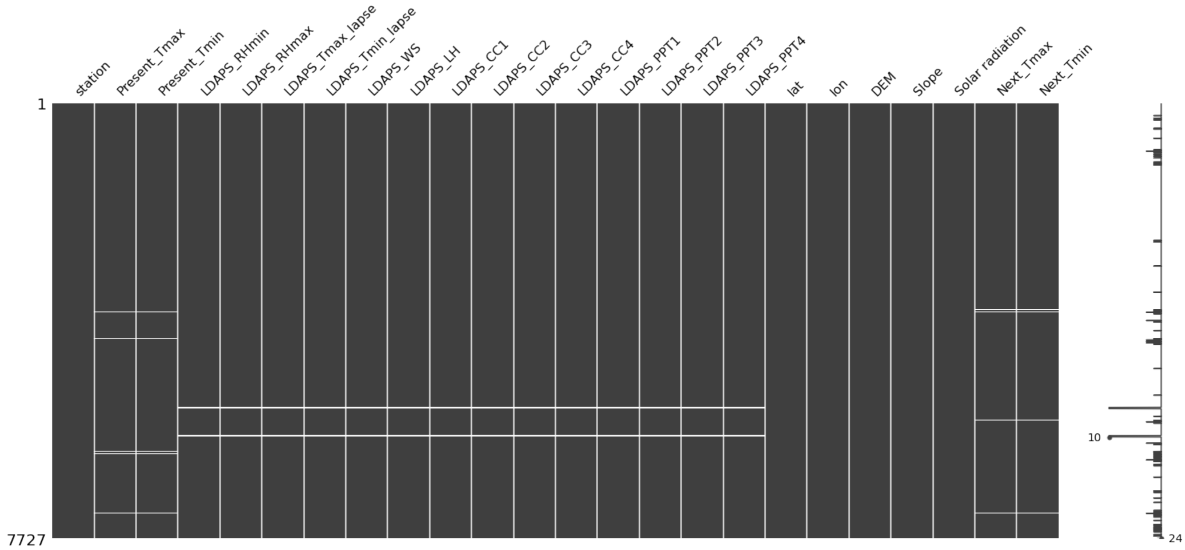


Figure 15: Visualization of the original dataset. White lines indicate `NaN`s.

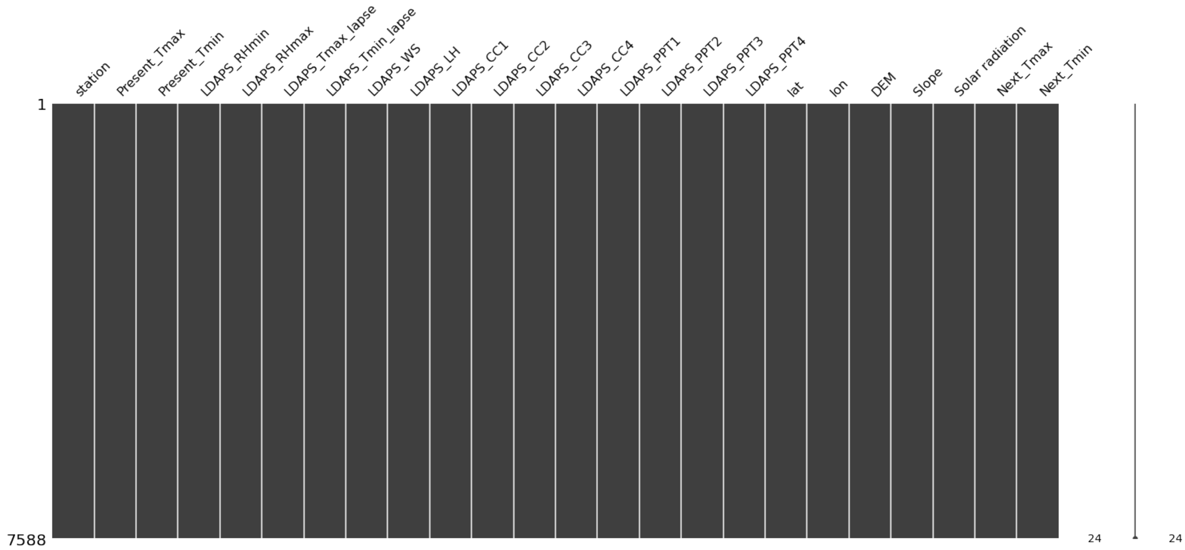


Figure 16: Visualization of the dataset after the removal of `NaN`s.

- 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.

3.3.1 No Regularization

The hyperparameter - number of clusters was sweeped and the value that resulted in the lowest validation SSE was chosen. The following cluster numbers were swept for: [1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100].

3.3.1.1 Predicting: `Next_Tmin`

The E_{rms} 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:

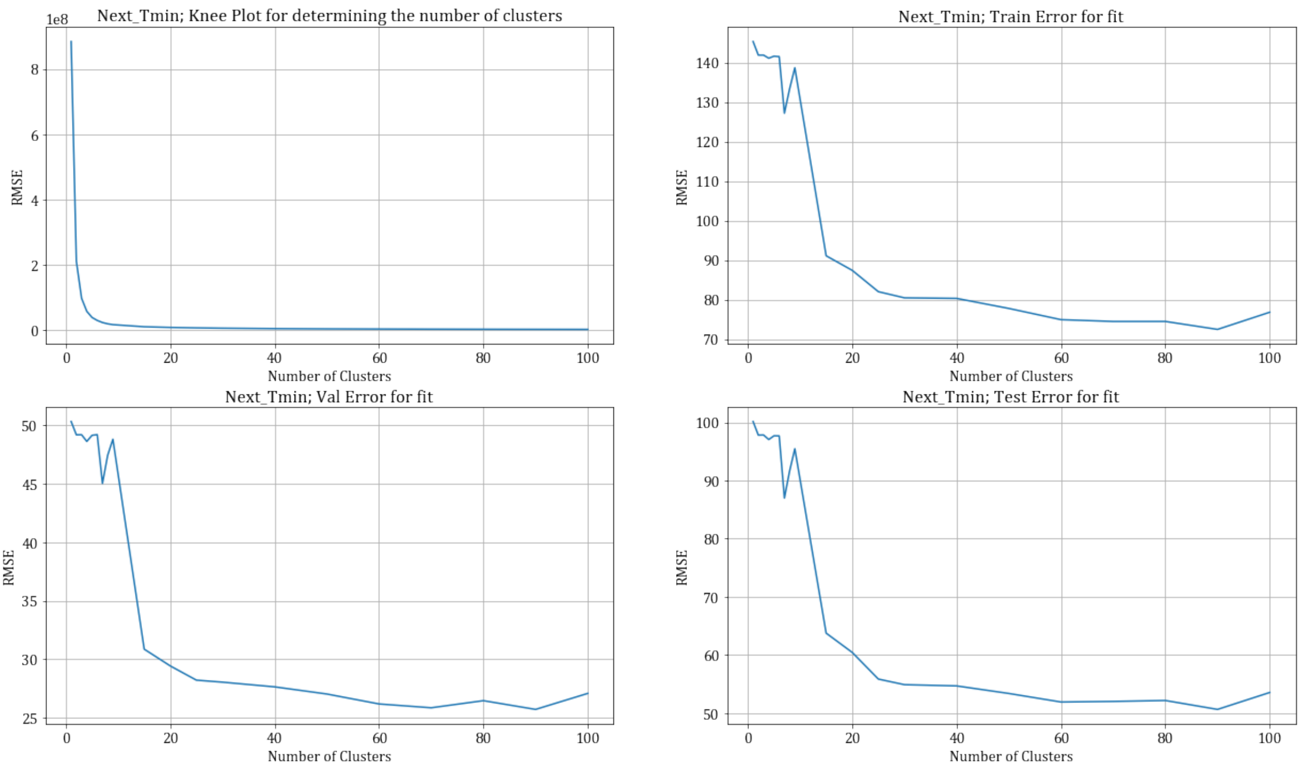


Figure 17: K-Means inertia, SSE on training and validation data from the left to right respectively.

The errors obtained in tabular format is as follows:

# Clusters	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
90	1.0480926961558428	1.1145328487370127	1.0612317092745545
70	1.077282613356503	1.1205646601791848	1.0900393186117472
60	1.0838258289557656	1.1347092563134056	1.0878633208966022
80	1.0773257656133501	1.1469280060128313	1.0933190935289967
50	1.124898166939697	1.171619621385478	1.1186915649161837

Table 9: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}.

The the number of clusters that resulted in the lowest RMSE is 90. In addition to the scatter plot, histograms were plotted to understand the variance in the data.

The histogram and scatter plot of the target and the model output is as follows (train data):

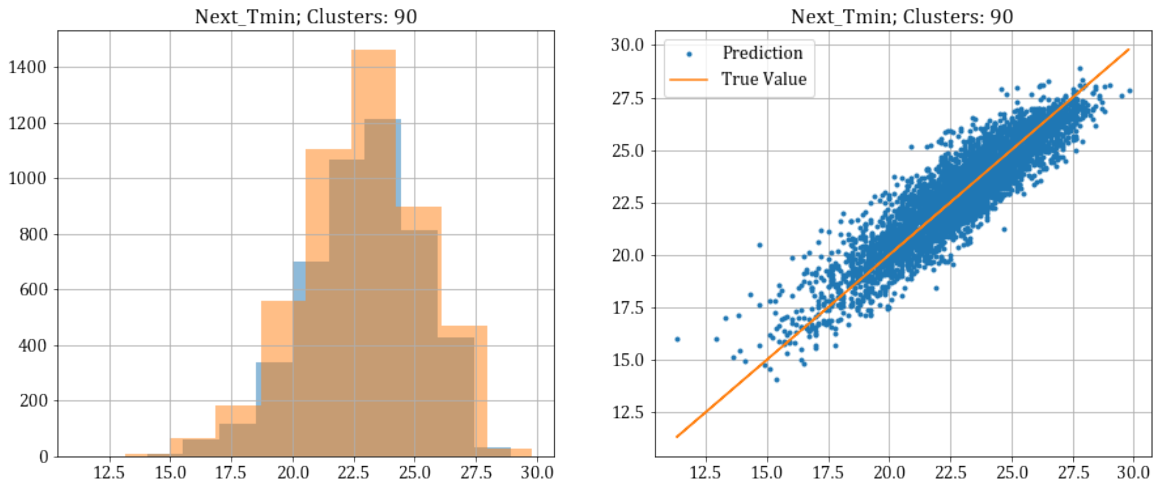


Figure 18: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0$

The histogram and scatter plot of the target and the model output is as follows (test data):

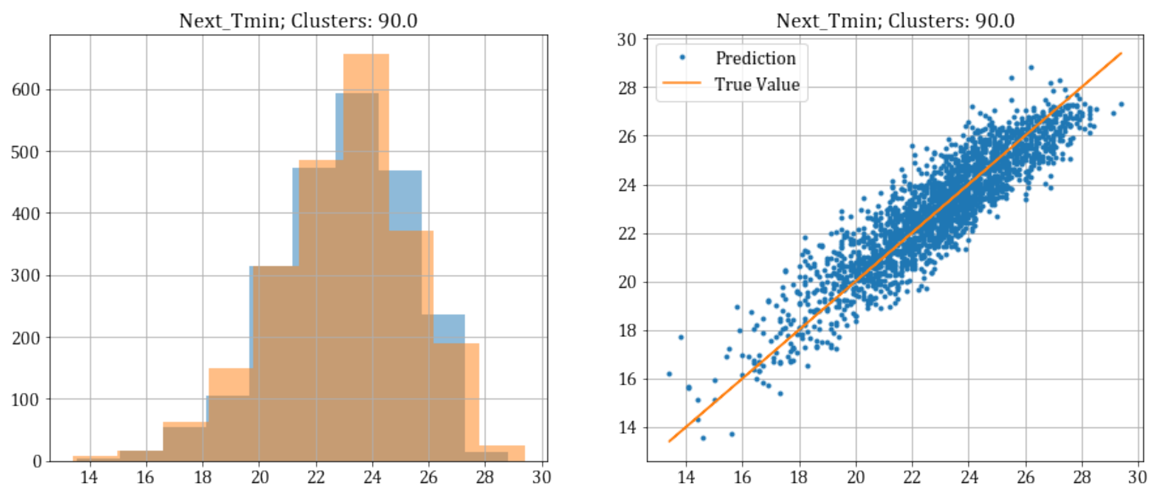


Figure 19: Histogram and Scatter plot of the target values against the model prediction for testing dataset, using linear regression with gaussian basis and $\lambda : 0$

The lowest Erms values obtained (number of clusters: 90) are as follows:

- Training E_{rms}: 1.0480926961558428
- Testing E_{rms}: 1.0612317092745545

3.3.1.2 Predicting: `Next_Tmax`

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

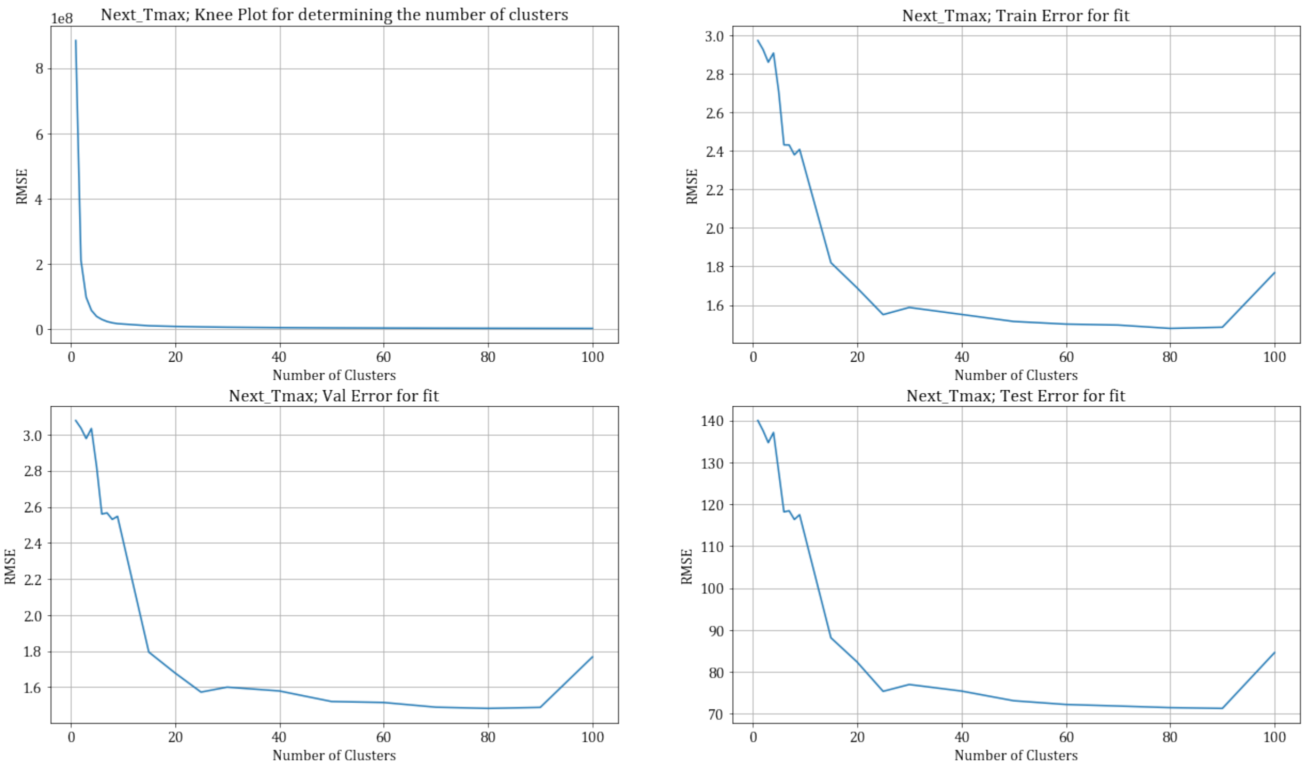


Figure 20: K-Means inertia, RMSE on training and validation data from the left to right respectively.

The errors obtained in tabular format is as follows:

# Clusters	E_{rms} Train	E_{rms} Validation	E_{rms} Test
80	1.4786466027685028	1.482350293596334	1.4982621391957776
90	1.484803258559306	1.4880444688982468	1.494427196289964
70	1.4960679385931273	1.4890949206178046	1.506337114473912
60	1.5009698930597348	1.5150336590549258	1.5139975307535112
50	1.5146941030926666	1.5205650411093423	1.5327334181249275

Table 10: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}.

The the number of clusters that resulted in the lowest RMSE is 80. In addition to the scatter plot, histograms were plotted to understand the variance in the data.

The histogram and scatter plot of the target and the model output is as follows (train data):

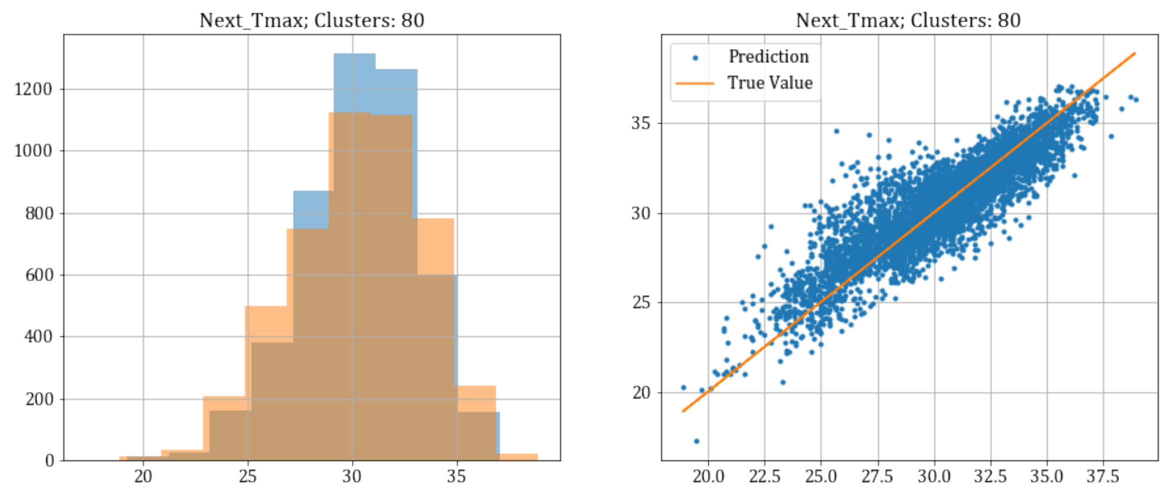


Figure 21: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0$

The histogram and scatter plot of the target and the model output is as follows (test data):

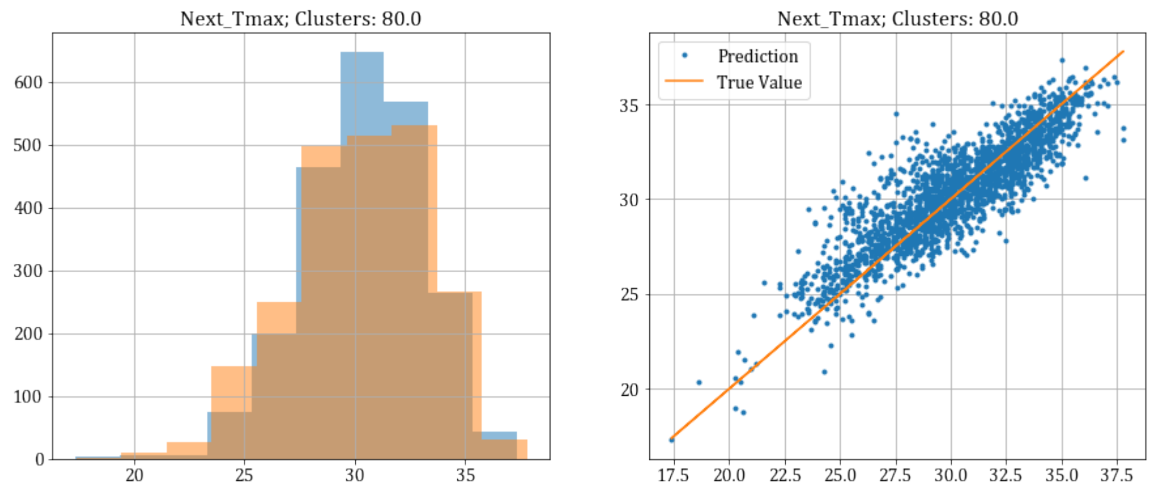


Figure 22: Histogram and Scatter plot of the target values against the model prediction for testing dataset, using linear regression with gaussian basis and $\lambda : 0$

The lowest E_{rms} values obtained (number of clusters: 80) are as follows:

- Training E_{rms}: 1.4786466027685028
- Testing E_{rms}: 1.4982621391957776

3.3.2 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 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

The hyperparameter - number of clusters and λ were sweeped and the value that resulted in the lowest validation SSE was chosen. The following cluster numbers were swept for: [1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100] and the following λ values were sweeped: [0.01, 0.1, 1.0, 5.0, 10.0].

3.3.2.1 Predicting: `Next_Tmin`

The Erms 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:

# Clusters	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
200	0.01	1.9013040128753353	1.9965582216927813	1.9047855943641792
190	0.01	1.904933411245413	1.9990574601548905	1.908742610287438
180	0.01	1.906872845452907	1.9994072596187717	1.9109215864242062
170	0.01	1.916154510430869	2.007065870193715	1.9205837284324376
160	0.01	1.9265592590375111	2.0159961584465105	1.9311538203795762

Table 11: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}.

The the number of clusters that resulted in the lowest RMSE is 200 and λ is 0.01. In addition to the scatter plot, histograms were plotted to understand the variance in the data.

The histogram and scatter plot of the target and the model output is as follows (train data):

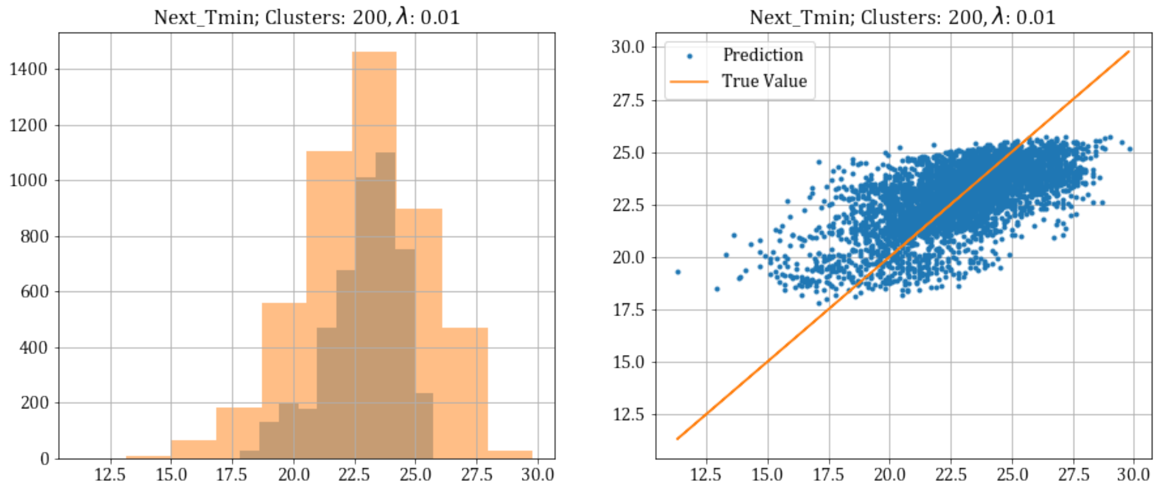


Figure 23: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0.01$

The histogram and scatter plot of the target and the model output is as follows (test data):

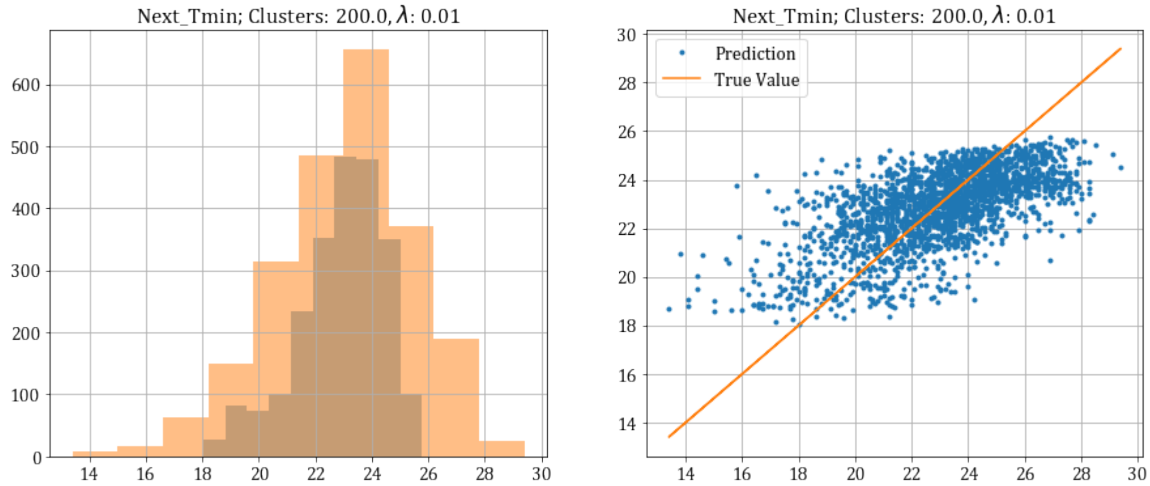


Figure 24: Histogram and Scatter plot of the target values against the model prediction for testing dataset, using linear regression with gaussian basis and $\lambda : 0.01$

The lowest Erms values obtained are as follows:

- Training E_{rms}: 1.9013040128753353
- Testing E_{rms}: 1.9047855943641792

3.3.2.2 Predicting: `Next_Tmax`

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

# Clusters	λ	E_{rms} Train	E_{rms} Validation	E_{rms} Test
200	0.01	2.249936156869166	2.399288774682482	2.302560104383557
190	0.01	2.2523172411109824	2.4004581548123065	2.3045855076045143
180	0.01	2.2602647948690384	2.4079528711983964	2.313110392501987
170	0.01	2.2640439964624384	2.4103143785285273	2.3164194038569397
160	0.01	2.2690106265072445	2.414998730544044	2.32150961586746

Table 12: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}.

The the number of clusters that resulted in the lowest RMSE is 200 and λ is 0.01. In addition to the scatter plot, histograms were plotted to understand the variance in the data.

The histogram and scatter plot of the target and the model output is as follows (train data):

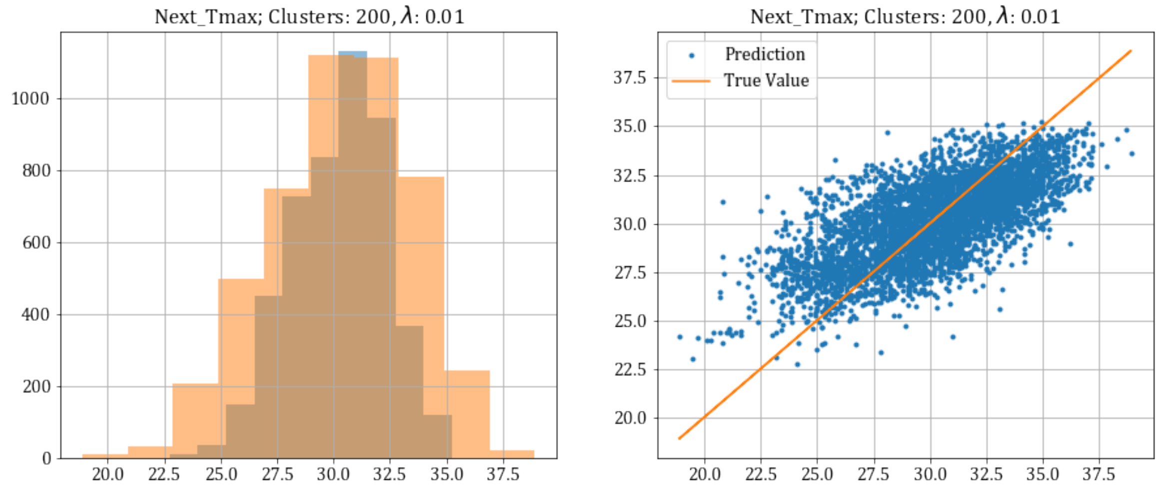


Figure 25: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0.01$

The histogram and scatter plot of the target and the model output is as follows (test data):

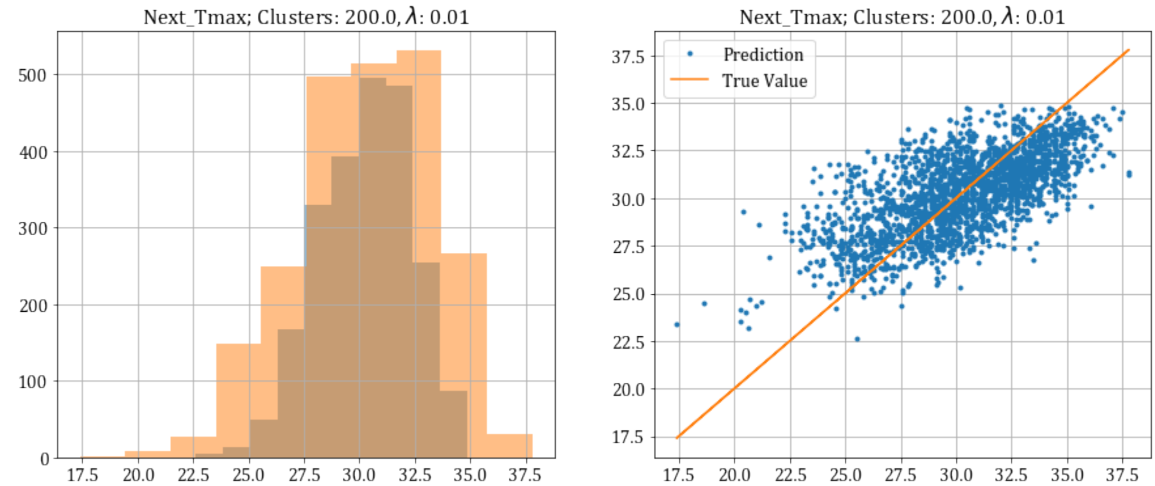


Figure 26: Histogram and Scatter plot of the target values against the model prediction for testing dataset, using linear regression with gaussian basis and $\lambda : 0.01$

The lowest Erms values obtained are as follows:

- Training E_{rms}: 2.2499361568691\u200966
- Testing E_{rms}: 2.302560104383557

3.3.3 Tikhonov Regularization

As defined in section 3.1, tikhonov regularization was applied to data set 3. Using the regularization coefficient and number of clusters as hyperparameters, gridsearch was done on the train, validation and test set and the model performing best on validation set was chosen.

3.3.3.1 Predicting: `Next_Tmin`

The following table gives the Erms using different combinations of hyperparameter values:

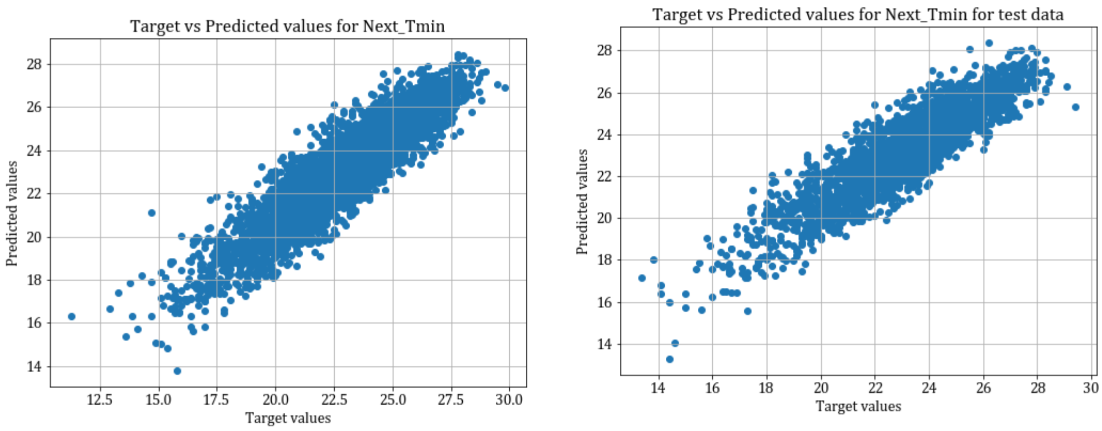


Figure 27: Gaussian basis model on dataset3 using tikhonov regularization for `Next_Tmin`

# Clusters	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
90	10.0	1.038	1.102	1.063
80	5.0	1.049	1.11	1.066
80	0.1	1.054	1.111	1.068
90	0.01	1.055	1.112	1.083
70	1.0	1.06	1.113	1.068
100	1.0	1.065	1.114	1.077
70	0.01	1.061	1.114	1.069
70	5.0	1.059	1.116	1.068
90	5.0	1.073	1.116	1.095

Table 13: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}, using Tikhonov regularization for `Next_Tmin`.

As seen in the table, the optimum value of the hyperparameters for `Next_Tmin` is **90** clusters and λ value of **10**. Using these values on the training and test data gives us the scatter plots in Figure 27 The lowest Erms values obtained are as follows:

- Training E_{rms}: 1.038
- Testing E_{rms}: 1.063

3.3.3.2 Predicting: `Next_Tmax`

The following table gives the Erms using different combinations of hyperparameter values:

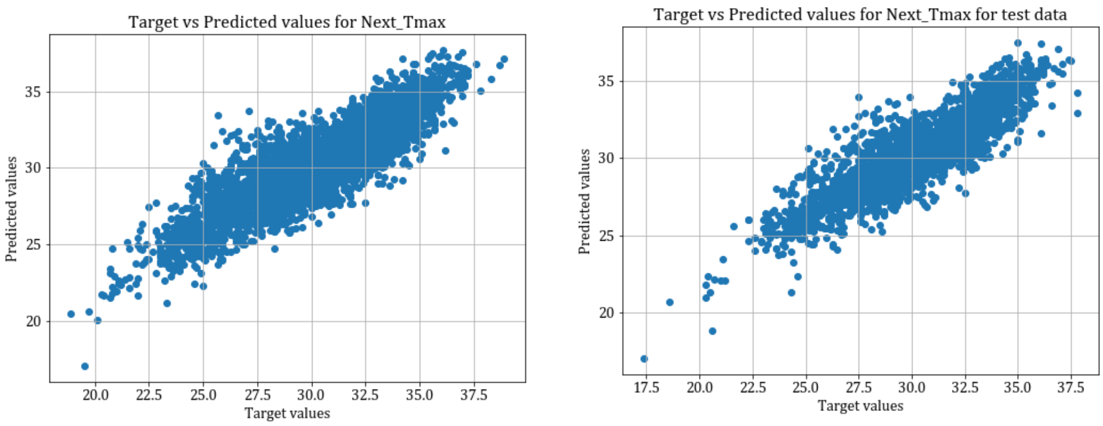


Figure 28: Gaussian basis model on dataset3 using tikhonov regularization for `Next_Tmax`

# Clusters	λ	E_{rms} **Train**	E_{rms} **Validation**	E_{rms} **Test**
90	0.01	1.454	1.467	1.474
90	5.0	1.444	1.469	1.466
80	0.1	1.473	1.477	1.491
80	5.0	1.48	1.479	1.505
90	10.0	1.464	1.482	1.497
80	0.01	1.475	1.483	1.504
90	1.0	1.47	1.484	1.497
70	5.0	1.472	1.484	1.493
70	0.1	1.468	1.485	1.49
70	0.01	1.477	1.486	1.497

Table 14: E_{rms} on the Training, Validation and Testing dataset, across different number of clusters, for 5 hyper parameters that result in the lowest E_{rms}, using Tikhonov regularization for `Next_Tmax`.

As seen in the table, the optimum value of the hyperparameters for `Next_Tmax` is **90** clusters and λ value of **0.01**. Using these values on the training and test data gives us the scatter plots in Figure 28

The lowest Erms values obtained are as follows:

- Training E_{rms}: 1.454
- Testing E_{rms}: 1.474

3.3.4 Inference

From the above plots we observe that:

- The model predictions are better when Tikhonov regularization is applied. The E_{rms} when quadratic regularization is applied is marginally higher than when λ is 0.
- The predictions of the model is better when the number of clusters considered is larger and hence, the E_{rms} is smaller.

- In case of no-regularization, we observed that the training, validation and testing E_{rms} increased at the highest number of clusters considered. This could be potentially due to improper initialization.
- In addition we see that the variance in the prediction is lower when the number of clusters considered is lower. This can be observed in the plots below:

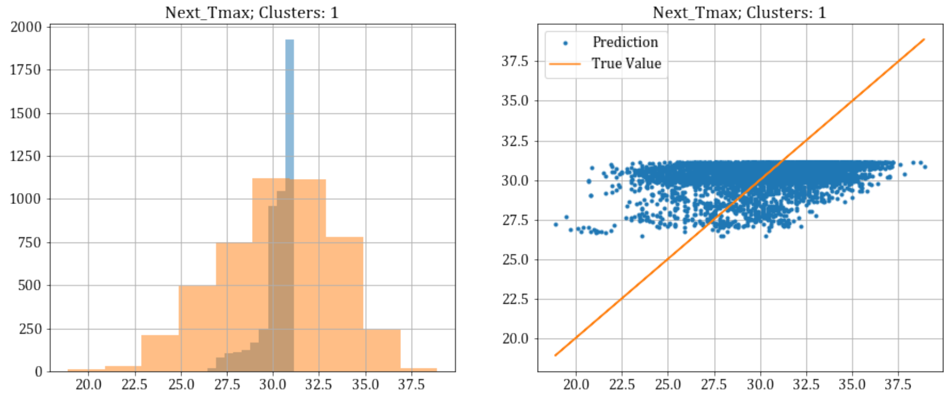


Figure 29: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0$

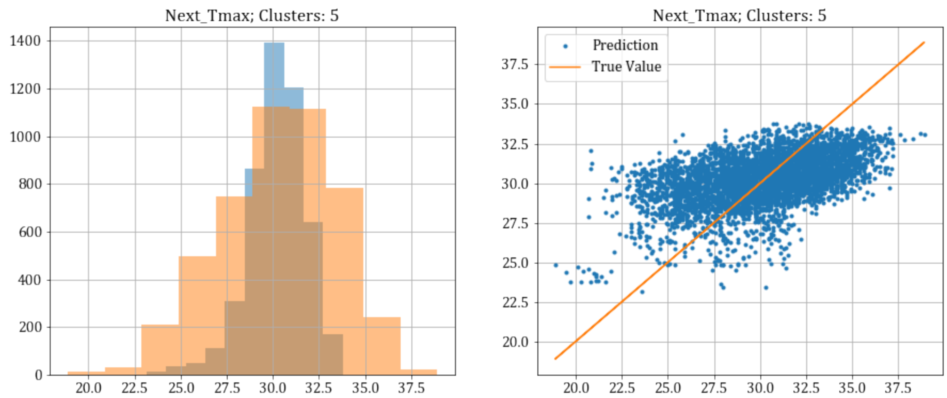


Figure 30: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0$

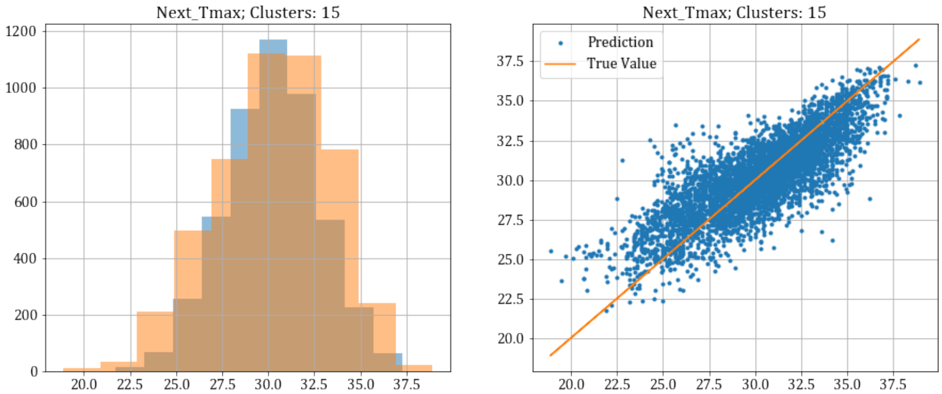


Figure 31: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0$

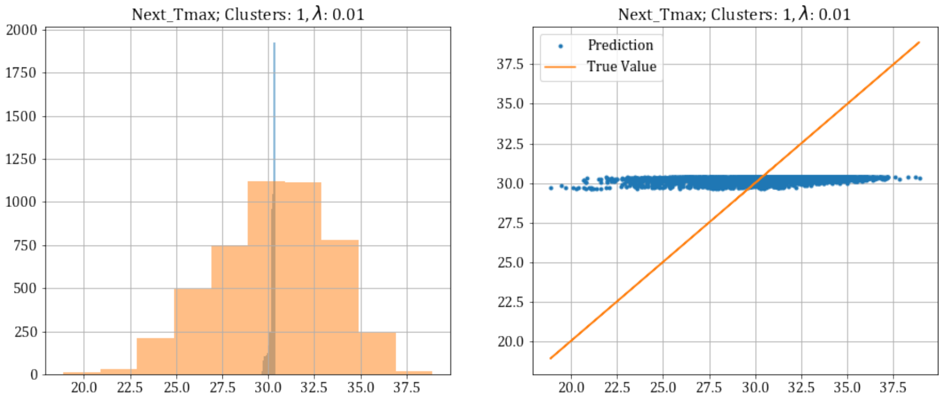


Figure 32: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0.1$

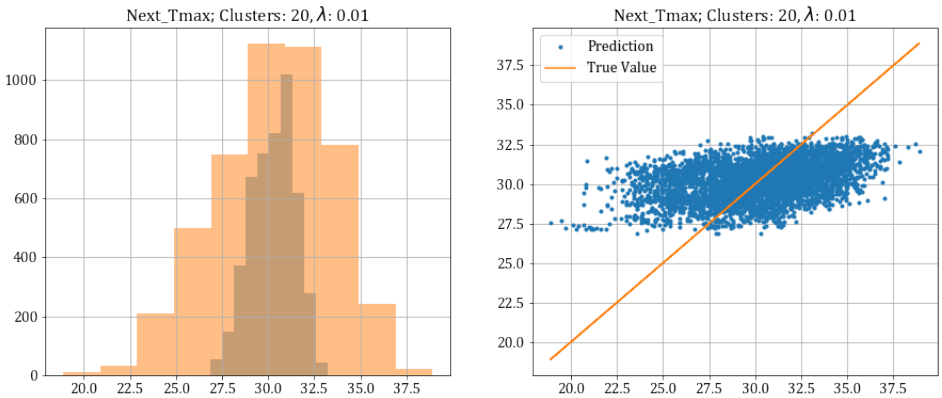


Figure 33: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0.1$

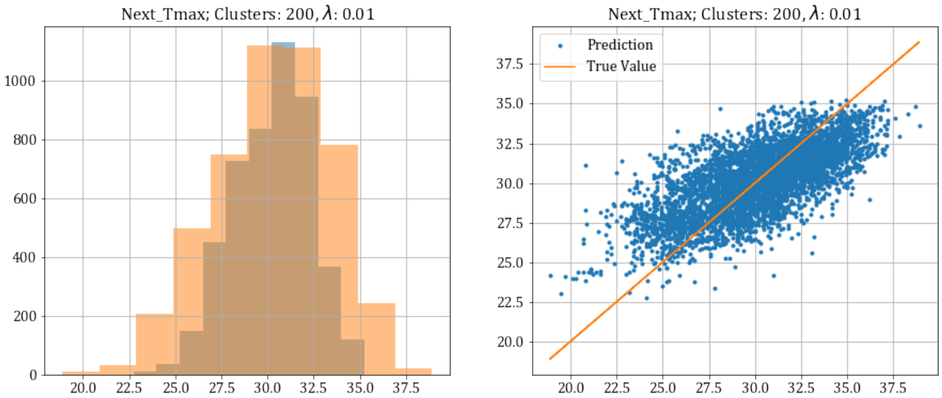


Figure 34: Histogram and Scatter plot of the target values against the model prediction for training dataset, using linear regression with gaussian basis and $\lambda : 0.1$