CS F320: Foundations of Data Science
Assignment 1

Prepared by:

Vaibhav Mishra ID: 2019A3PS1350H
Avinash Gondela ID: 2019A8PS1357H
Akhilesh Senapati ID: 2019AAPS1352H

Contents:

Part A

Statement: Tabulate the minimum training and testing error achieved by your model by using polynomials of degree 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 to predict the output. Visualize the surface plots of your predictions (using matplotlib and Axes3D) that you obtained by using polynomials of varying degree and comment on how overfitting actually works.

Part A.) Tabulate the minimum training and testing error achieved by your model by using polynomials of degrees 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 to predict the output.
>>
Training and testing error for polynomials of degree 0 through 9
The proposed coefficient for GDA for degree zero polynomial is 1.4456930121525847e-16 (~0)
RMSE for degree zero for GDA is 0.9999999999999999

For degrees 1 to 9 for Gradient Descent Algorithm

Polynomial Degree	Training Error	Testing Error
1	1.06625	1.05389
2	1.15288	1.13108
3	1.47902	1.4304
4	1.73848	1.64277
5	1.41976	1.33837
6	1.30267	1.22955
7	1.41246	1.40011
8	1.53096	1.51773
9	1.48802	1.50228

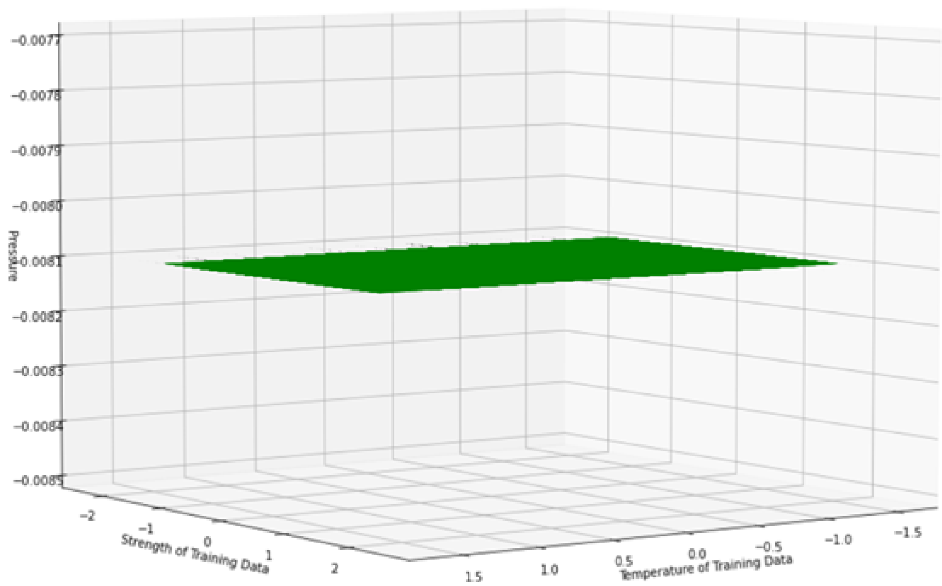
The proposed coefficient for degree zero in SGDA is -0.00810047591341151
RMSE for degree zero for SDA is 0.9669739704427083

For degrees 1 to 9 for Stochastic Gradient Descent Algorithm

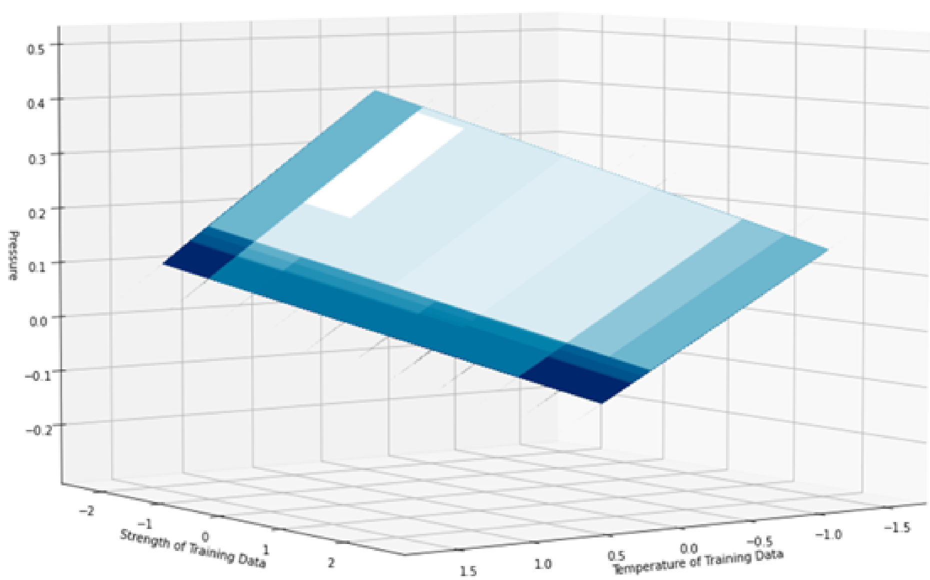
Polynomial Degree	Training Error	Testing Error
1	1.04345	1.09836
2	1.14887	1.16799
3	1.64998	1.48411
4	2.60465	1.72296
5	5.01916	1.38776
6	9.29304	1.36364
7	15.3749	1.49426
8	16.9224	1.45736
9	14.8352	1.38913

Surface Plots for Stochastic Gradient Descent Algorithm for degree 0 to 9

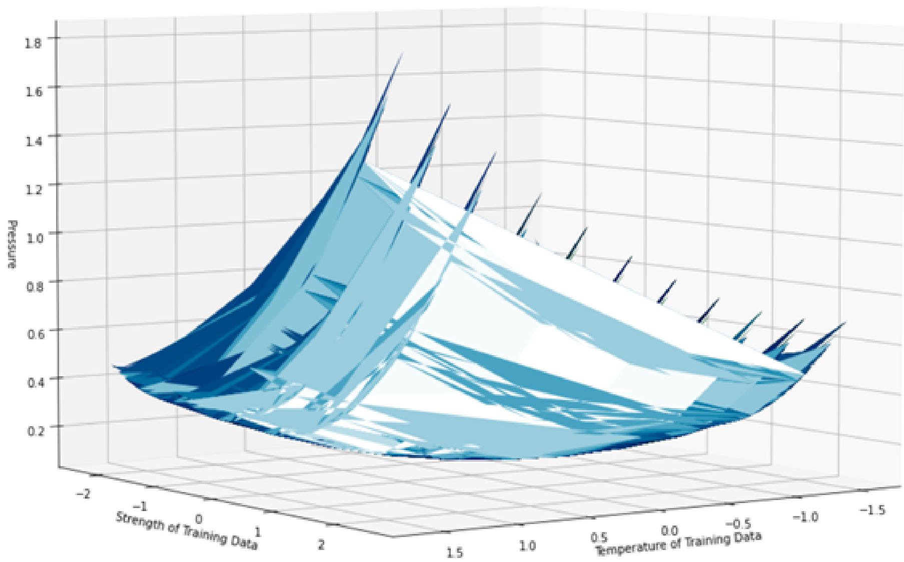
For a polynomial of degree 0 in Stochastic Gradient Descent Algorithm surface plot is as follows



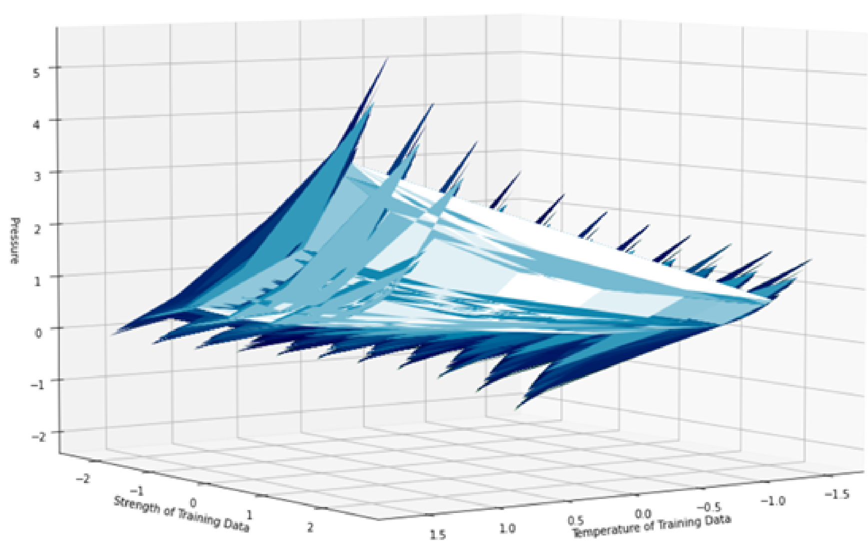
For polynomial of degree 1 in Stochastic Gradient Descent Algorithm surface plot is as follows



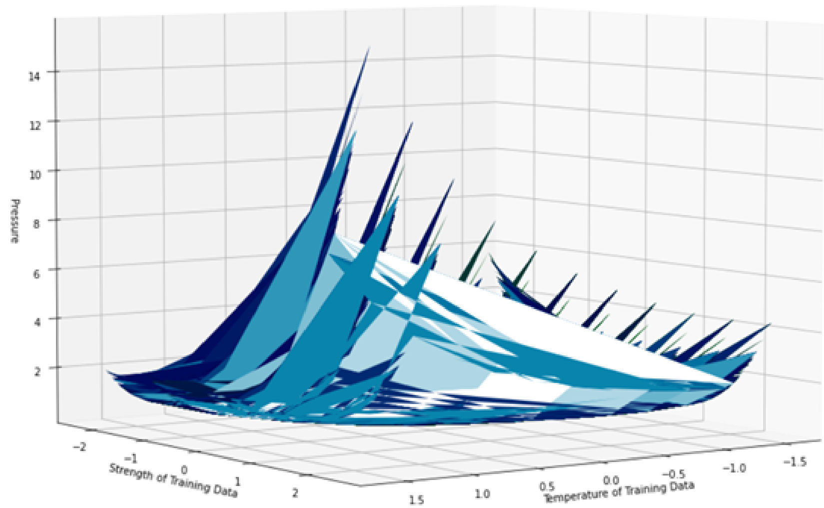
For polynomial of degree 2 in Stochastic Gradient Descent Algorithm surface plot is as follows



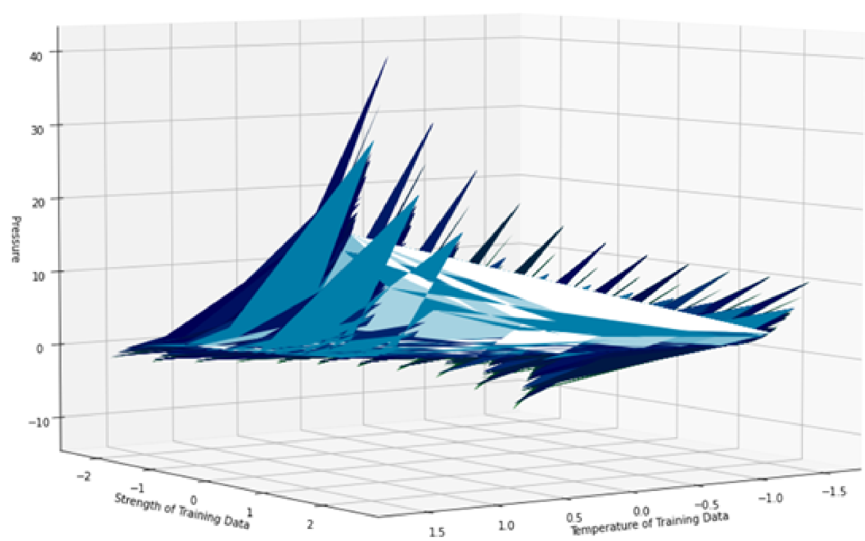
For polynomial of degree 3 in Stochastic Gradient Descent Algorithm surface plot is as follows



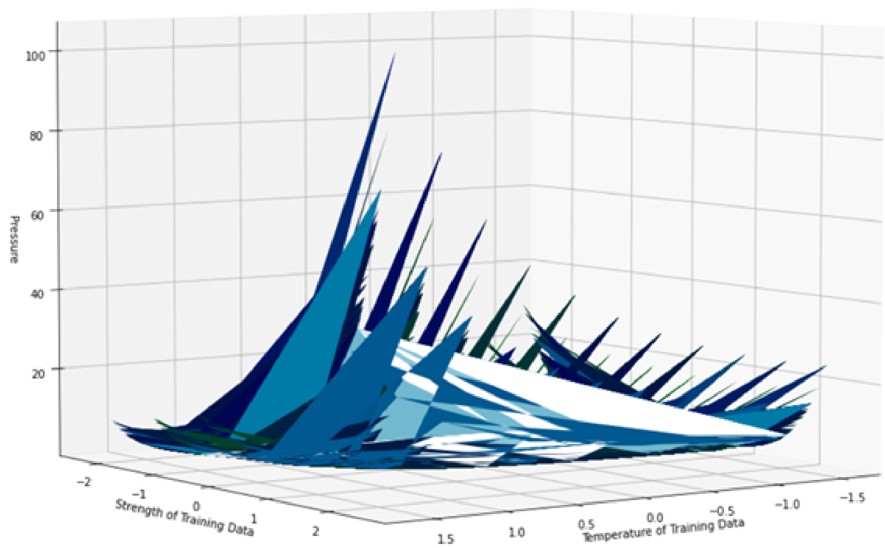
For polynomial of degree 4 in Stochastic Gradient Descent Algorithm surface plot is as follows



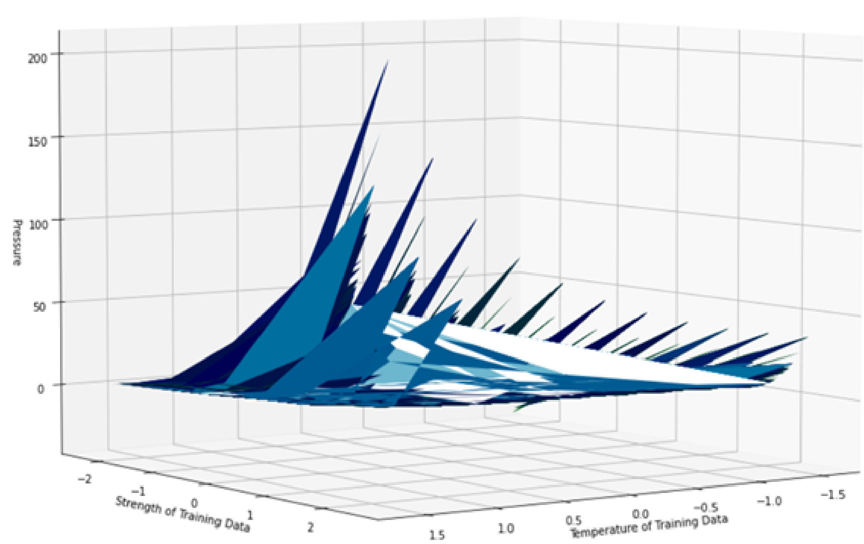
For polynomial of degree 5 in Stochastic Gradient Descent Algorithm surface plot is as follows



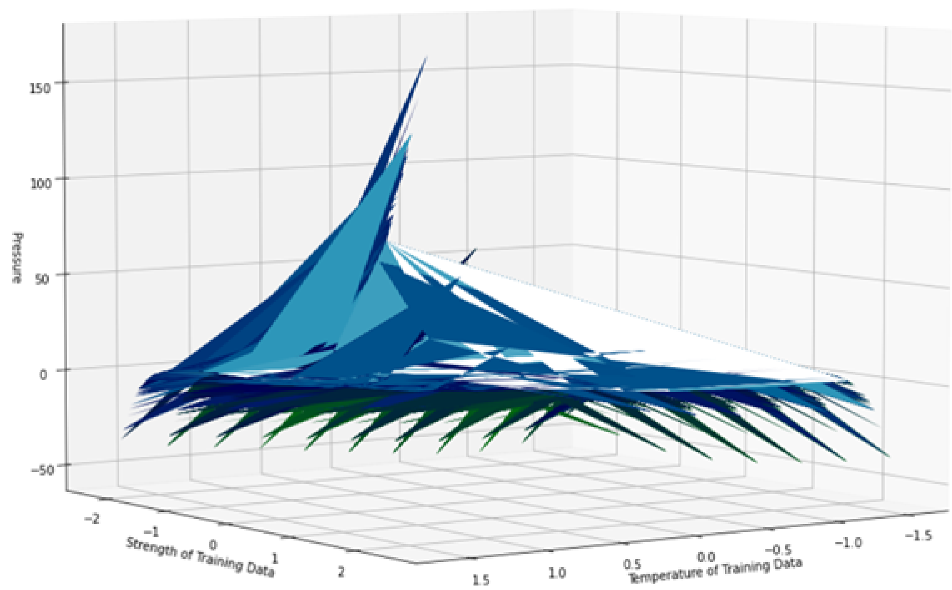
For polynomial of degree 6 in Stochastic Gradient Descent Algorithm surface plot is as follows



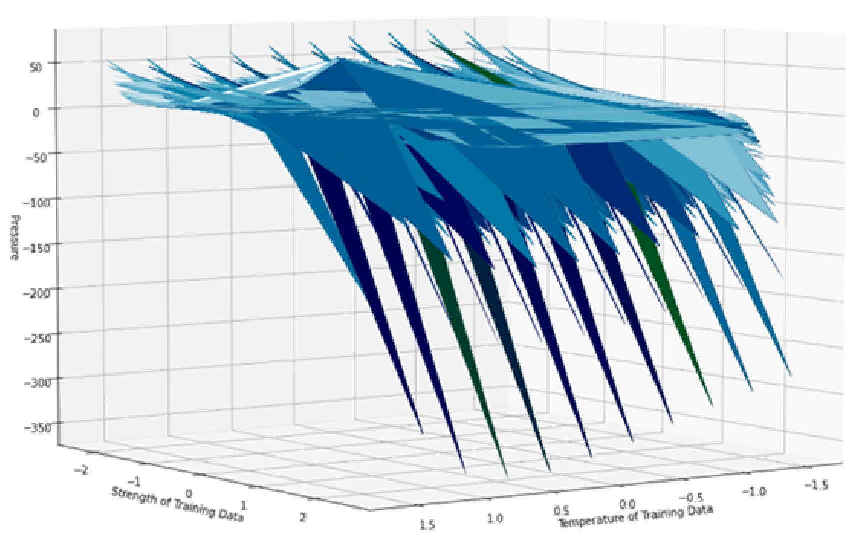
For polynomial of degree 7 in Stochastic Gradient Descent Algorithm surface plot is as follows



For polynomial of degree 8 in Stochastic Gradient Descent Algorithm surface plot is as follows

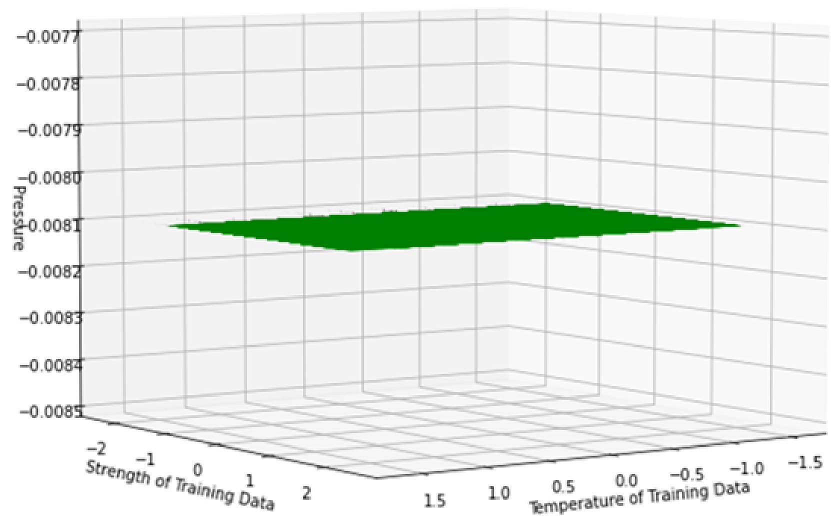


For polynomial of degree 9 in Stochastic Gradient Descent Algorithm surface plot is as follows

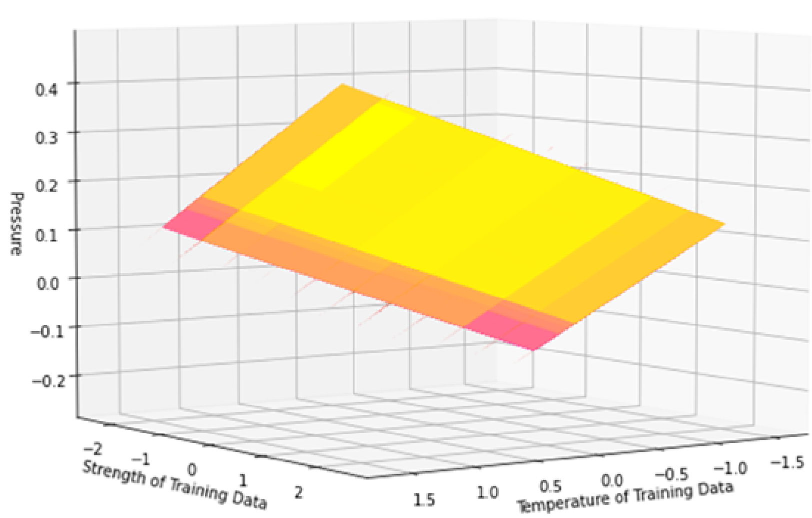


Now plotting Surface Plots for polynomial of Degree 0 to 9 for Gradient Descent Algorithm

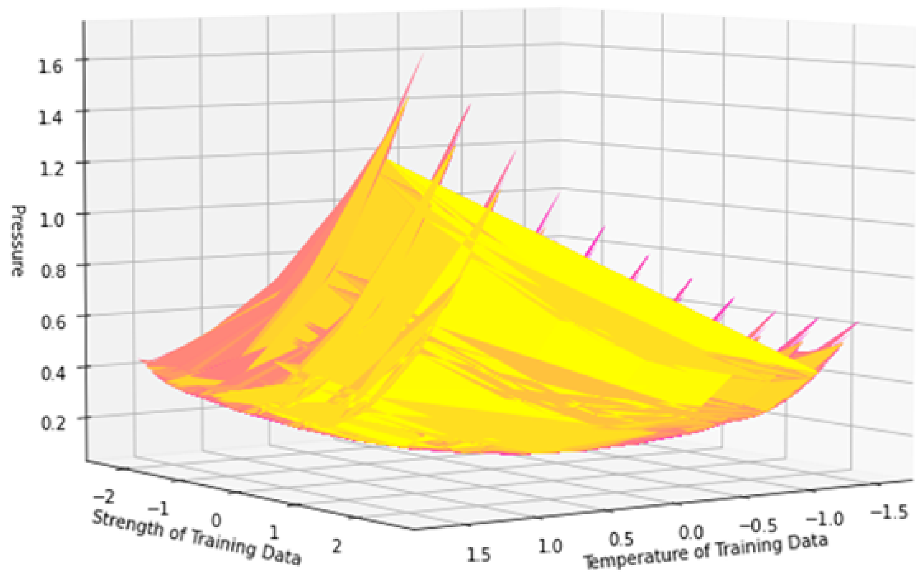
For polynomial of degree 0 in Gradient Descent Algorithm surface plot is as follows



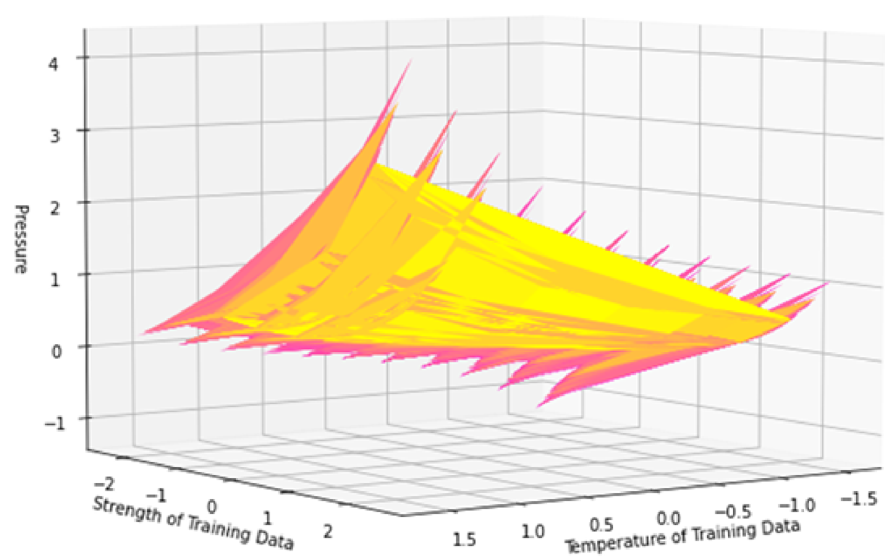
For polynomial of degree 1 in Gradient Descent Algorithm surface plot is as follows



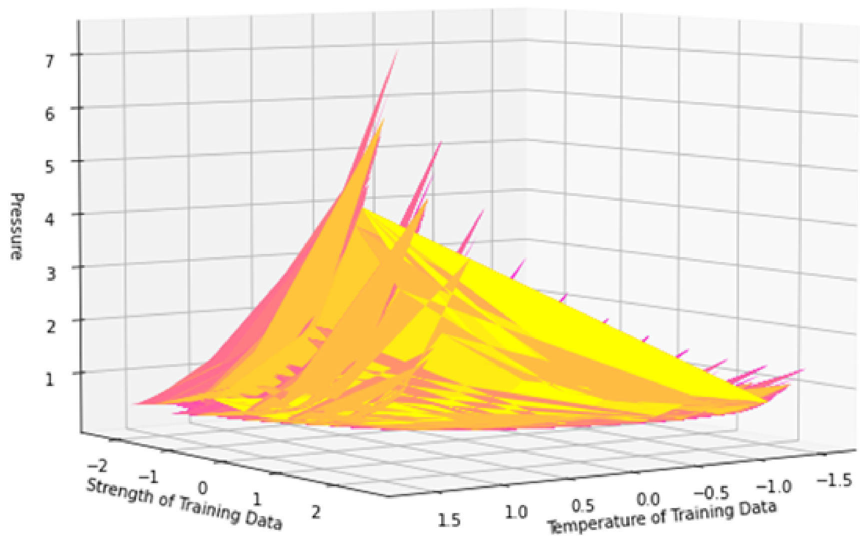
For polynomial of degree 2 in Gradient Descent Algorithm surface plot is as follows



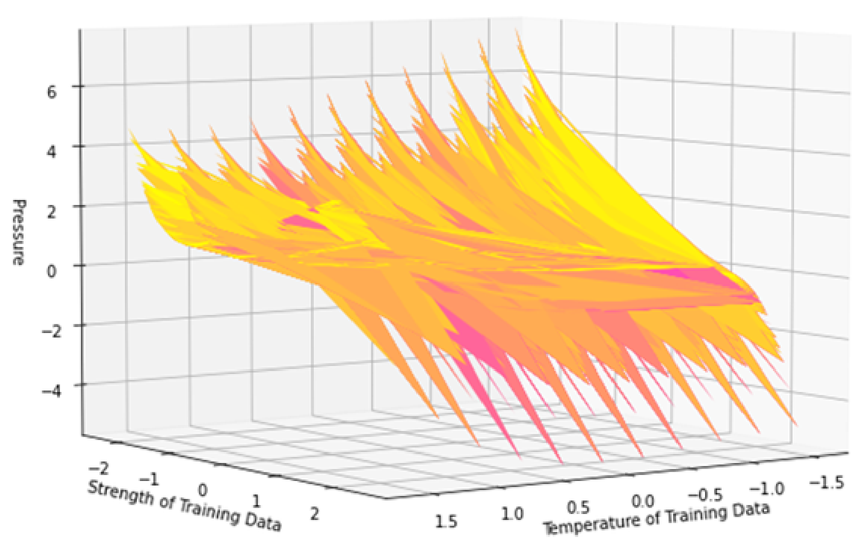
For polynomial of degree 3 in Gradient Descent Algorithm surface plot is as follows



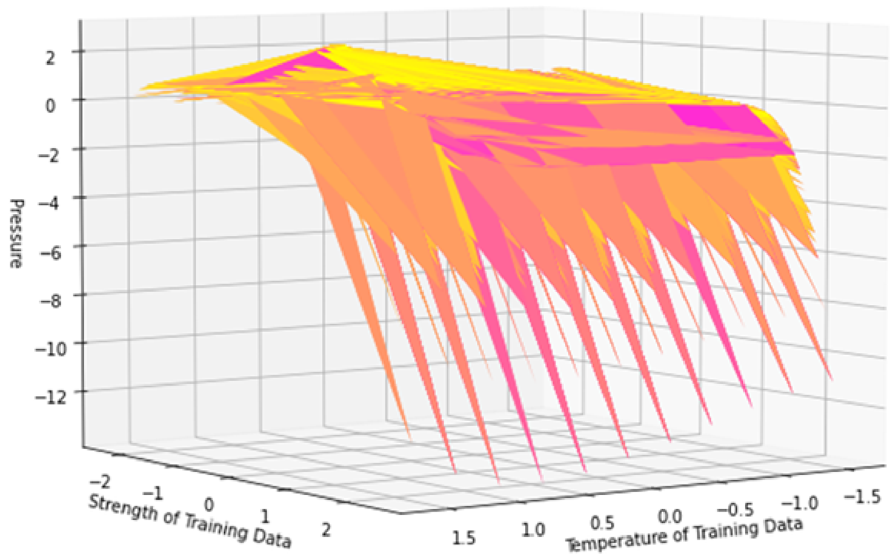
For polynomial of degree 4 in Gradient Descent Algorithm surface plot is as follows



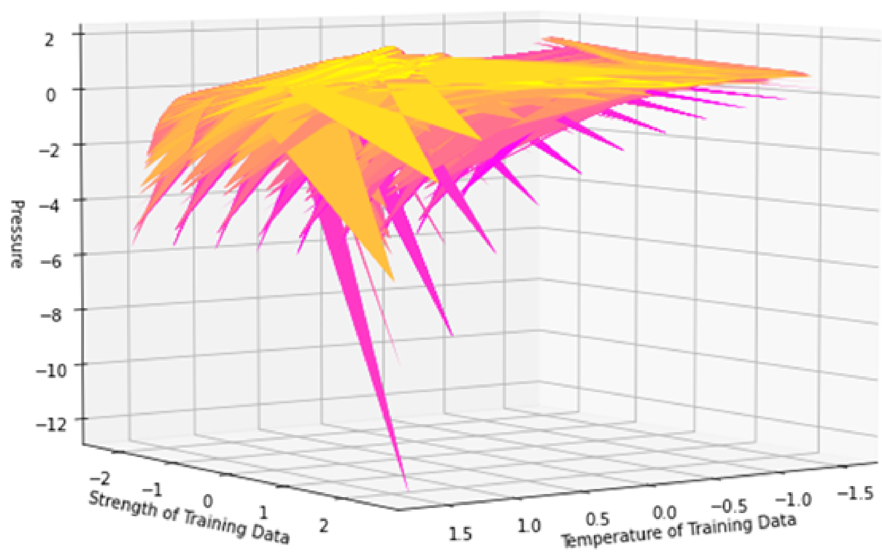
For polynomial of degree 5 in Gradient Descent Algorithm surface plot is as follows



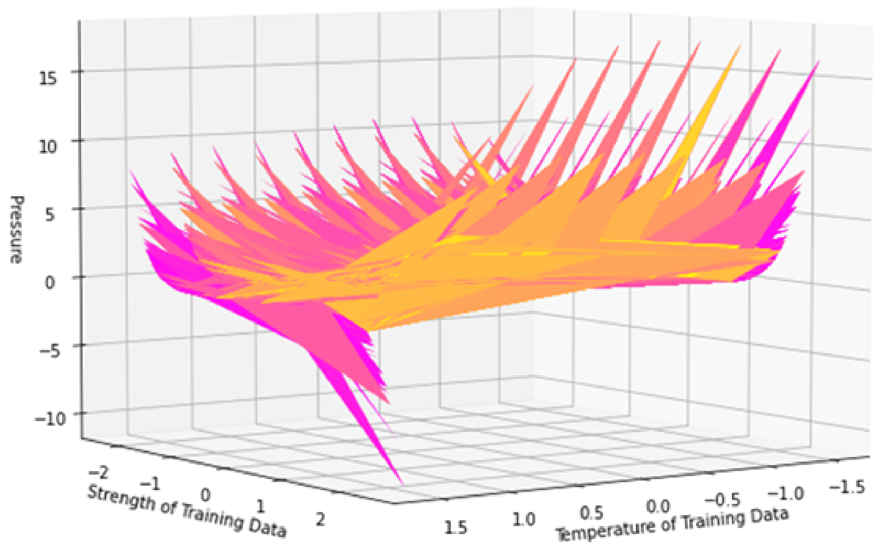
For polynomial of degree 6 in Gradient Descent Algorithm surface plot is as follows



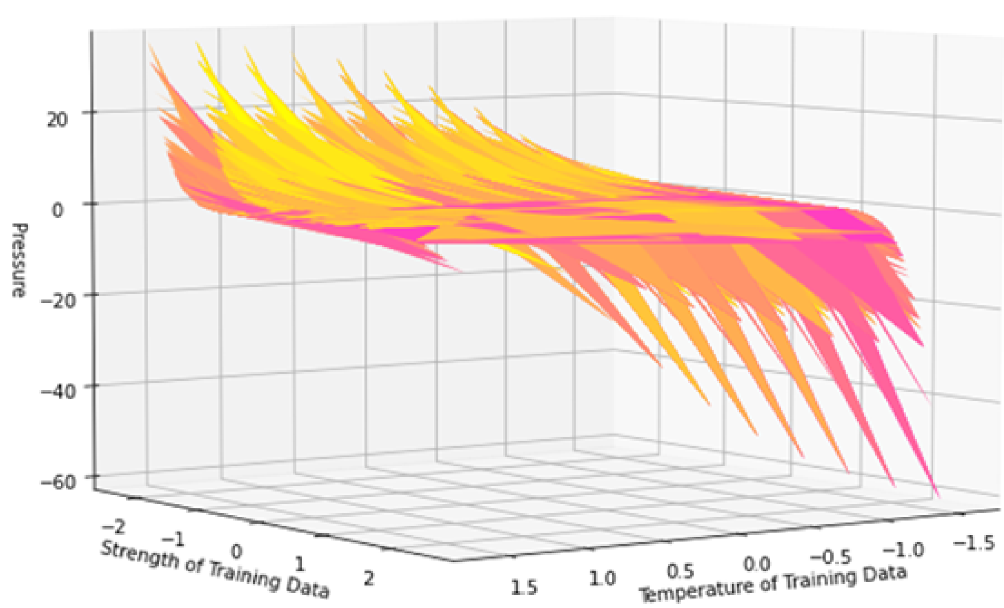
For polynomial of degree 7 in Gradient Descent Algorithm surface plot is as follows



For polynomial of degree 8 in Gradient Descent Algorithm surface plot is as follows



For polynomial of degree 9 in Gradient Descent Algorithm surface plot is as follows



Q.)Comment on how overfitting works?

>> In overfitting the model tries to accommodate for all the training data points for subsequent corrections. It overgeneralizes the model for the training data and may result in huge Root mean square error for testing data as it is not used for computation of the coefficients. Overfitting is characterised by low error rates and considerable variation. To avoid this, a portion of the training dataset is usually kept aside as a "test set" to ensure that no overfitting occurs. Overfitting occurs when the training data has a low error rate and the test data has a high error rate.

Part B

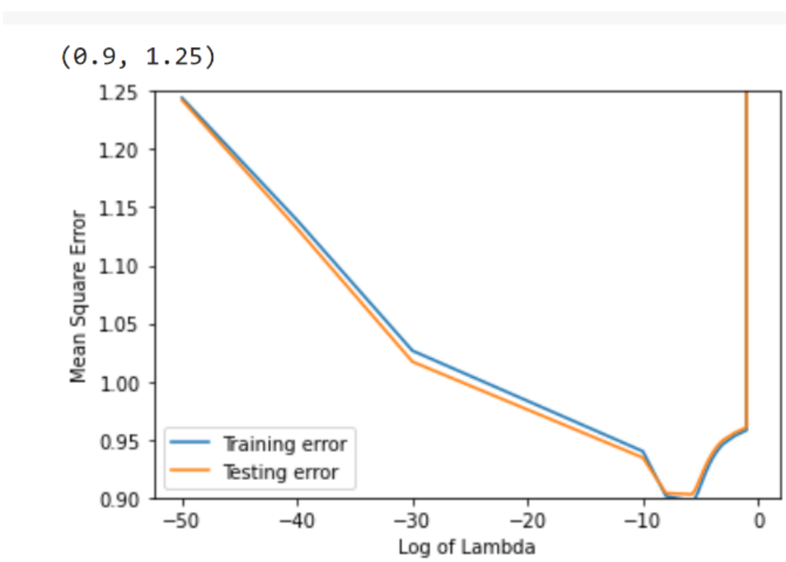
Statement: Tabulate the minimum training and testing error achieved by your model for 5 different values of lambda. Draw a plot of the root-mean square error vs the logarithm of lambda to figure out the optimal model.

Ridge Regression

Values:

Lambda Value	Training Data Error	Testing Data Error
exp(-8)	1.2443317460565293	1.242371396793574
exp(-7.5)	1.1387640332182565	1.1316422208709185
exp(-7)	1.026462124133172	1.0170022209684844
exp(-6.5)	0.9398077764106821	0.9343174681698055
exp(-6)	0.901304118837792	0.9038016137216603

Graph:



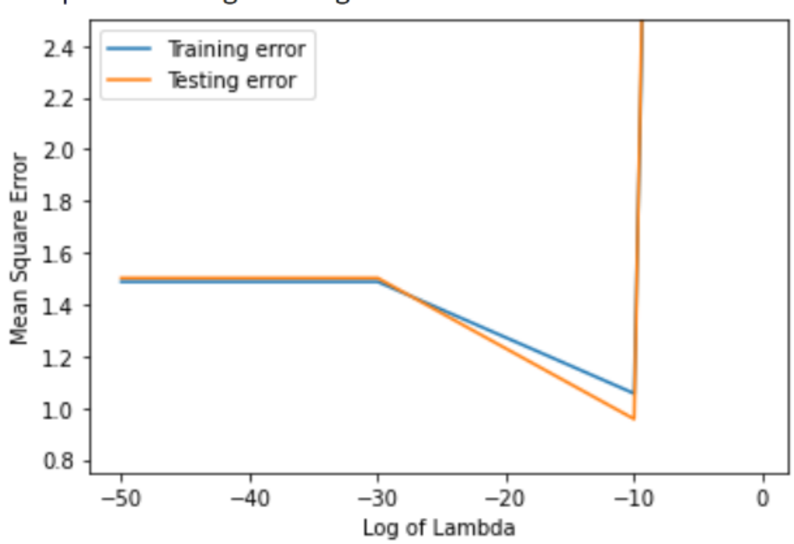
The most optimal value of lambda occurs at exp(-6.5) for Ridge regression. The same can be seen above in the graph. Around that lambda value the difference between the training and testing error is minimum and hence optimal.

Lasso Regression

Lambda Value	Training Error Data	Testing Error Data
exp(-50)	1.4880188045787015	1.5022759978486617
exp(-40)	1.4880188045786424	1.50227599784859
exp(-30)	1.4880188032737287	1.5022759962690857
exp(-10)	1.0557810978001676	0.9568077906001634
exp(-8)	5.579784946740305	6.141905153998064

Graph:

<matplotlib.legend.Legend at 0x7f80307d9290>



The most optimal value of lambda occurs at exp(-10) for Lasso regression. The same can be seen above in the graph. Around that lambda value the difference between the training and testing error is minimum and hence optimal.

Comparison

After comparing the models from both A and B,

The least error obtained in part A is:
Training Error: 1.06625
Testing Error: 1.05389

The least error obtained in part B is:
Training Error: 0.901304118837792
Testing Error: 0.9038016137216603

From the above results we can conclude that the best model obtained in Model B (Ridge Regression) has outperformed the best model obtained in Model A (Polynomial of Degree 1).