Polynomial Regression :

Polynomial regression is an algorithm developed to map the best fit relationship between feature tuples to the target attribute that are modelled in nth degree polynomials .(0 , 1, 2 , 3 , 4 etc….)

Gradient Descent algorithm :

An algorithm used to optimize the cost function(E) by updating the weight vector at each iteration so that error function (E) minimizes at each iteration and reaches a minimum value resulting in a best fit curve for our model.

$$W^{(k+1)}{}_{1} = W^{k}{}_{1} - \eta\left(\frac{\partial E}{\partial W_{1}}\right)_{W_{1}=W^{k}{}_{1}}$$

Stochastic Gradient Descent algorithm :

This algorithm uses the same formula or a technique to update the weight but in stochastic GD we do not modify the whole weight vector but rather keep on iterating a single weight from the vector which is more faster than GD and also gives the most appropriate model than GD and by solving normal equations.

Regularization:

It is a technique to fit the overfitting models as best as possible i.e, when a model of higher polynomial degree with less data points overfits (training error = 0), we try to constrain the weights so that the model doesn't overfit as its freedom of having random crusts and troughs
(overfitting) is reduced.

This is done by adding an extra term to the error function i.e,

$$E = \frac{1}{2} \left(\Sigma (Y - Y\hat{})^2 \right) + \lambda (\Sigma |w|)$$

- (LASSO REGRESSION)

$$E = \frac{1}{2} \left(\Sigma (Y - Y\hat{})^2 \right) + \lambda/2 (\Sigma |w|^2)$$

-(RIDGE REGRESSION

GRADIENT DESCENT ALGORITHM :(table of min errors of training and testing)

Degree	Minimum Training error	Minimum Testing error
0	0.9991341991341988	0.9788077668794599

1	0.2020153182992323	0.19101562342107625
2	0.2003995636123157	0.18866164455288834
3	0.18551885506607754	0.1772466520772907
4	0.1946346399000087	0.1841922604163948
5	0.19548612605400162	0.1861508810967773
6	0.2491083811774199	0.22593275196170878
7	0.41970040410638926	0.3805730913918861
8	0.5644964268083625	0.5201428186474416
9	0.6781612275135898	0.649695868233599

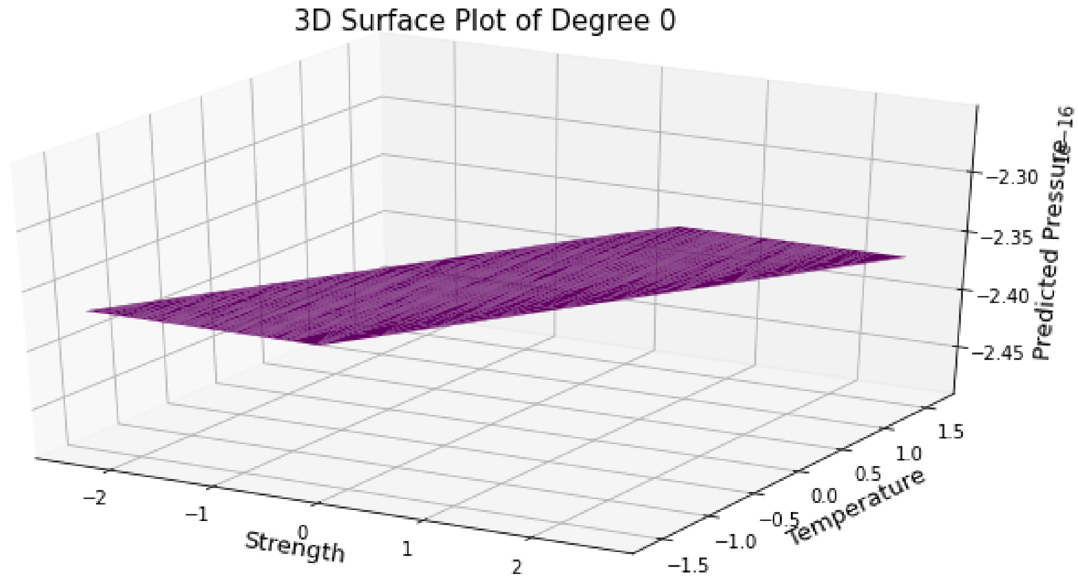
STOCHASTIC GRADIENT DESCENT ALGORITHM :(table of min errors of training and testing)

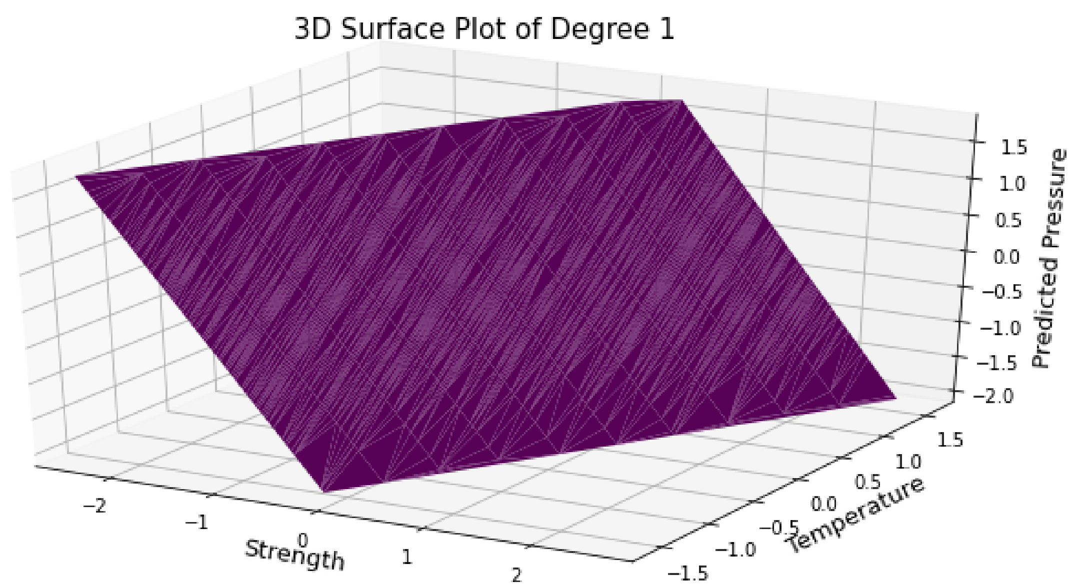
Degree	Minimum Training error	Minimum Testing error

0	1.0316499559323788	0.9946240286529372
1	0.20588705284116413	0.1914430202038772
2	0.21467360905227417	0.19801277881579674
3	0.18968174078799308	0.17988116663259424
4	0.208770300974709	0.19495370824258573
5	0.331419778410272	0.291651370318635
6	0.36221813749839277	0.32358455697461164
7	0.6979709819121396	0.6660073890071784
8	0.7820177966720819	0.7513595337671941
9	0.9078918018775062	0.9312078084369544

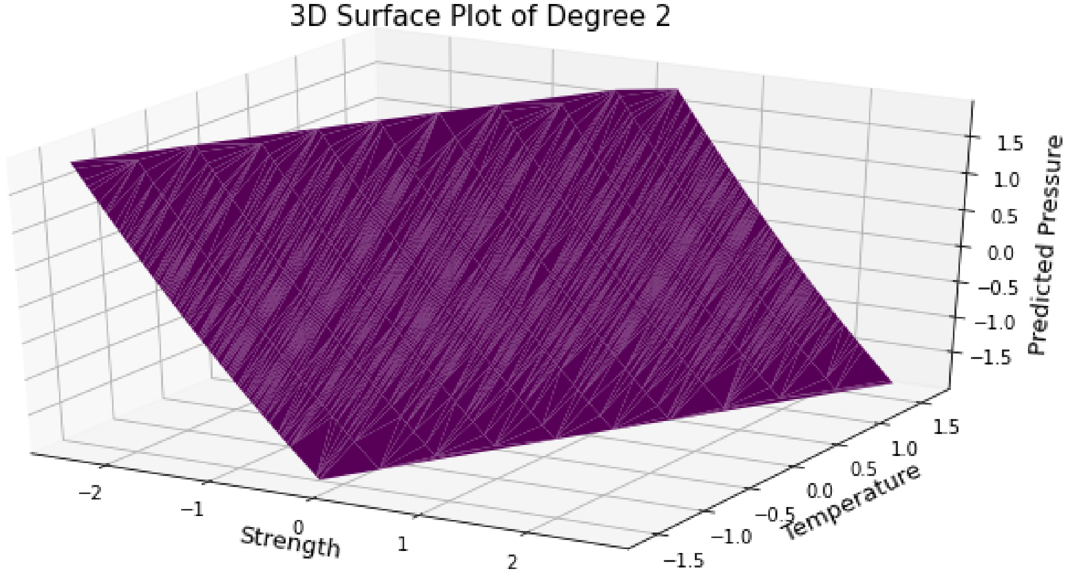
The surface plots of our prediction of different polynomial degrees using

GRADIENT DESCENT ALGORITHM :

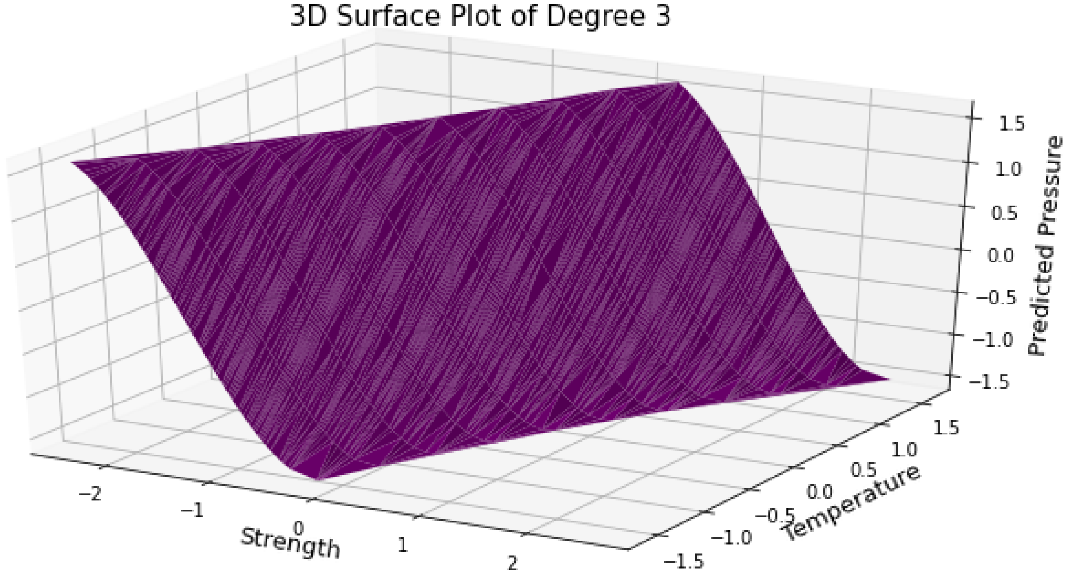




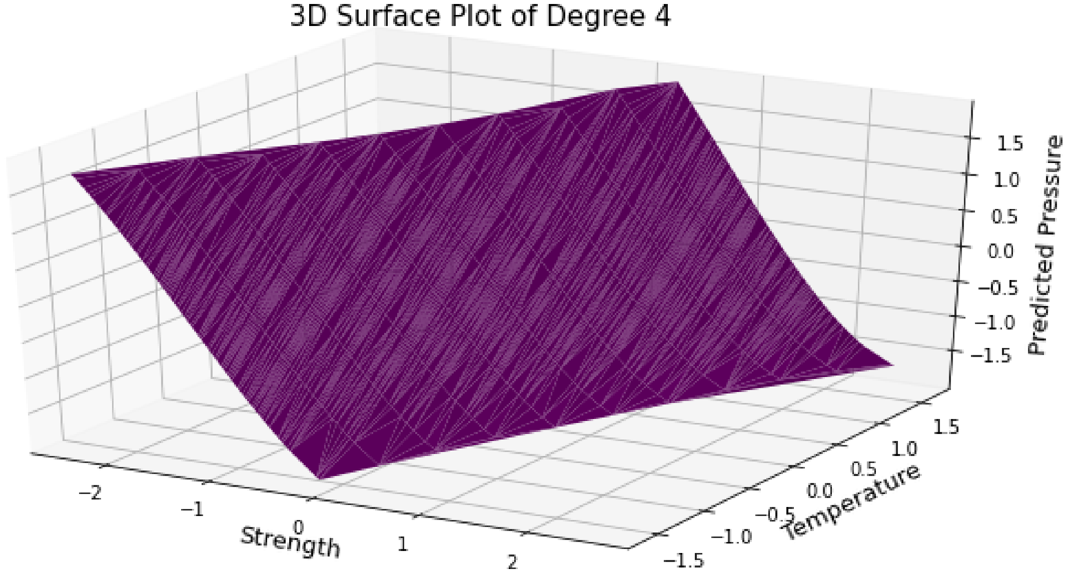
3D Surface Plot of Degree 2



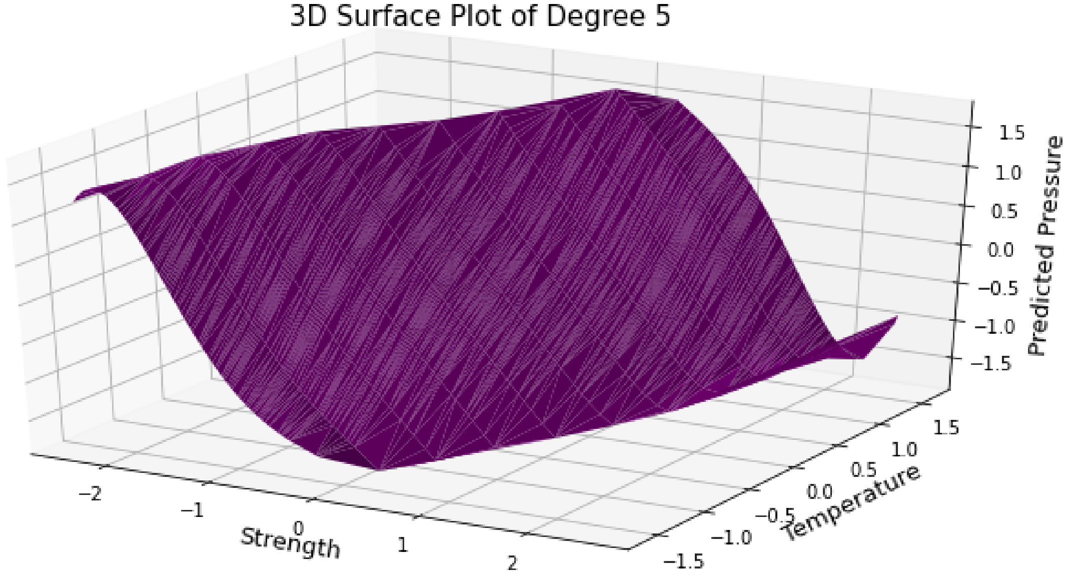
3D Surface Plot of Degree 3



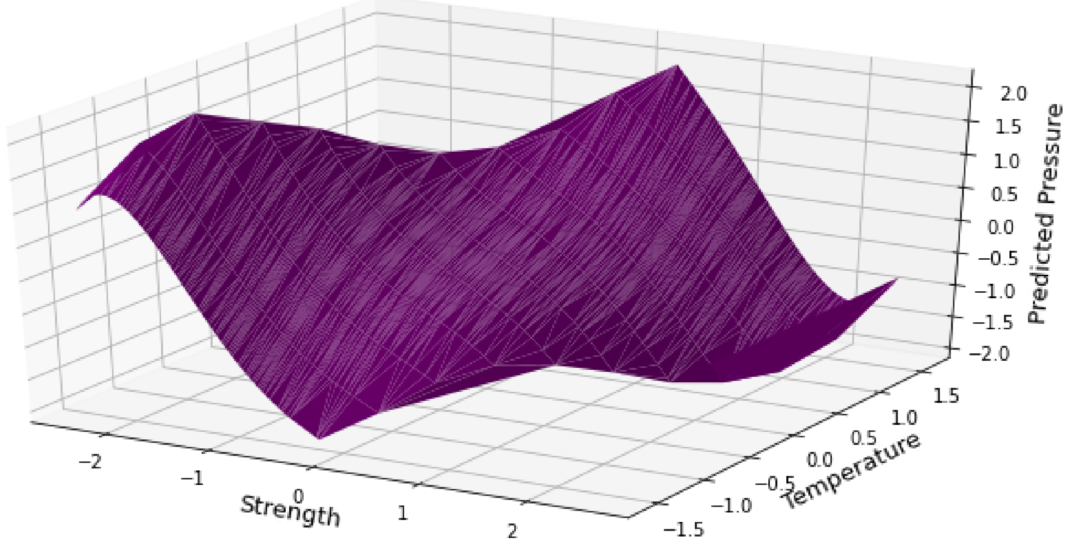
3D Surface Plot of Degree 4



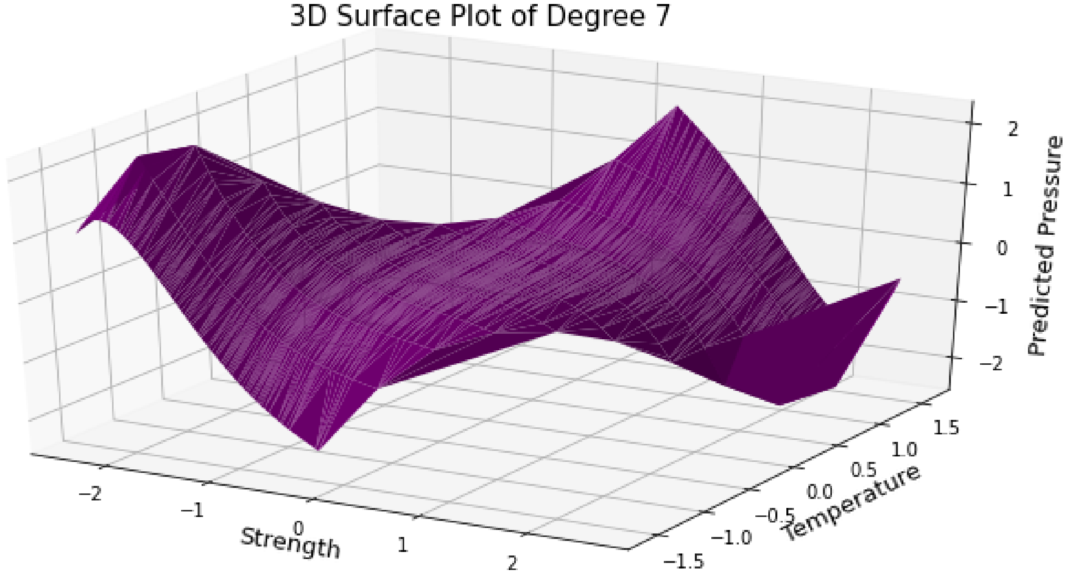
3D Surface Plot of Degree 5

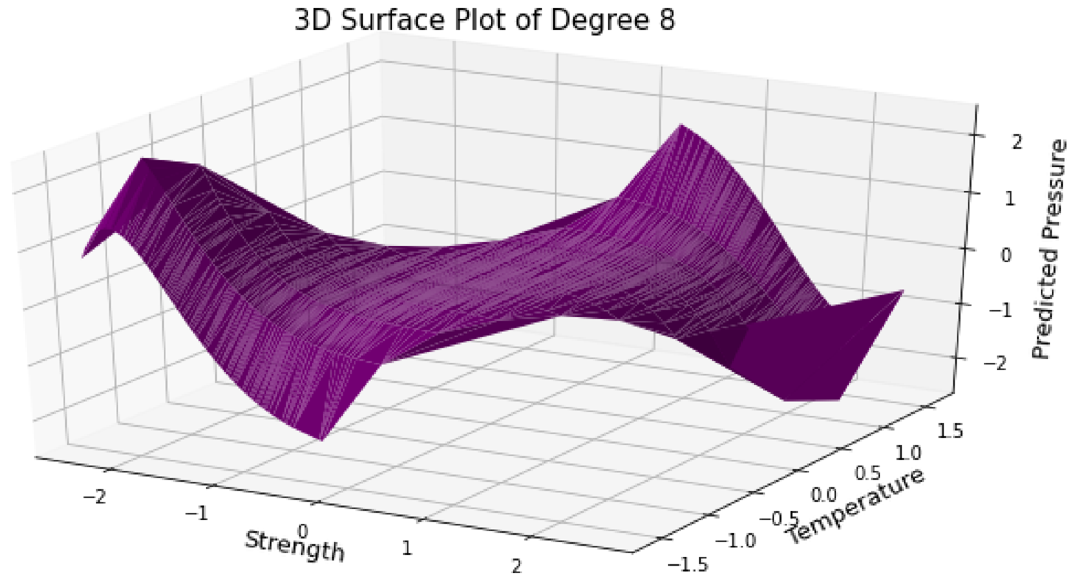


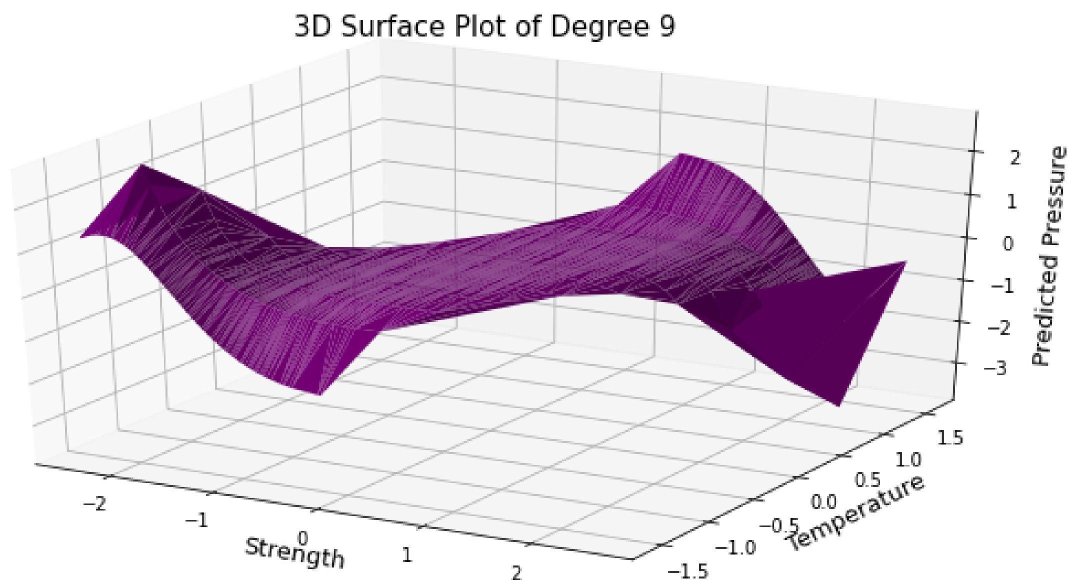
3D Surface Plot of Degree 6



3D Surface Plot of Degree 7

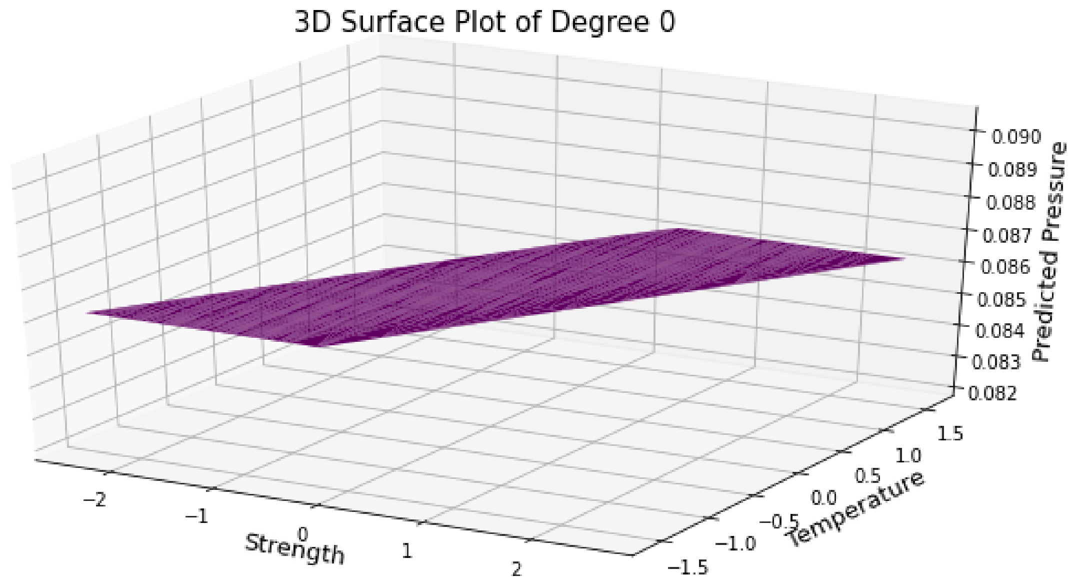


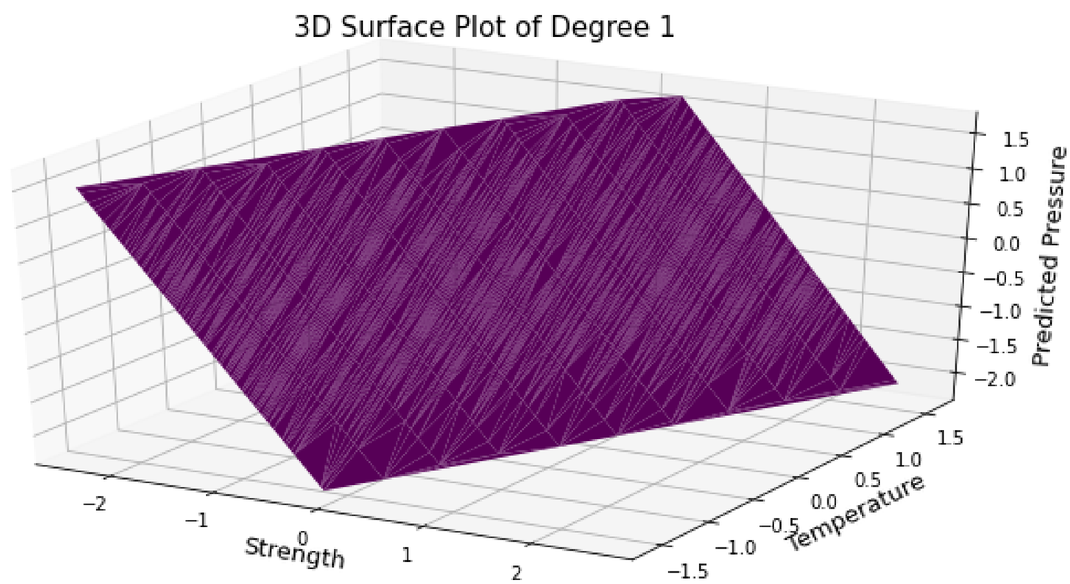

3D Surface Plot of Degree 8
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

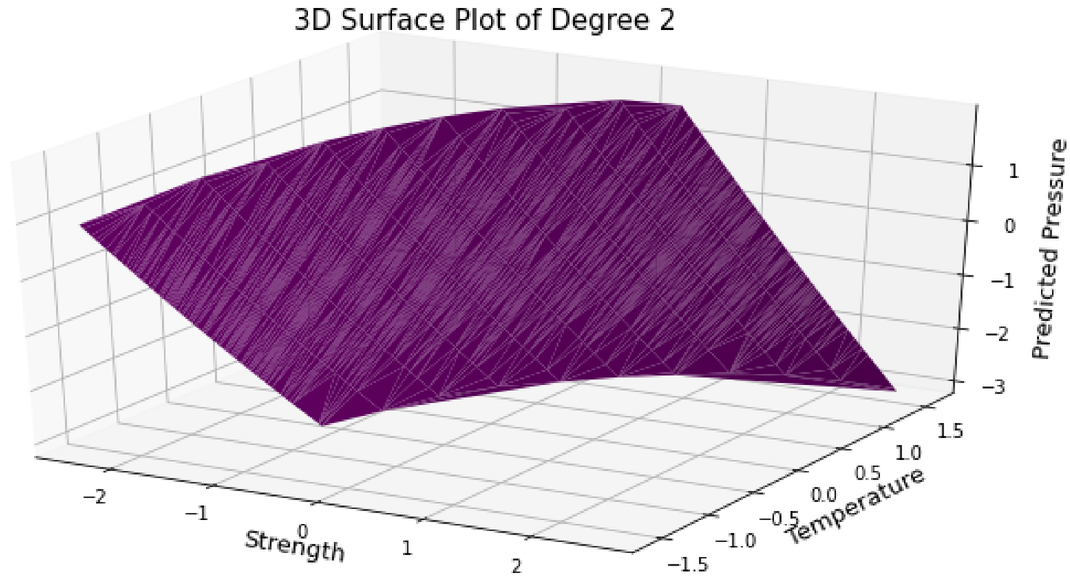

3D Surface Plot of Degree 9
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

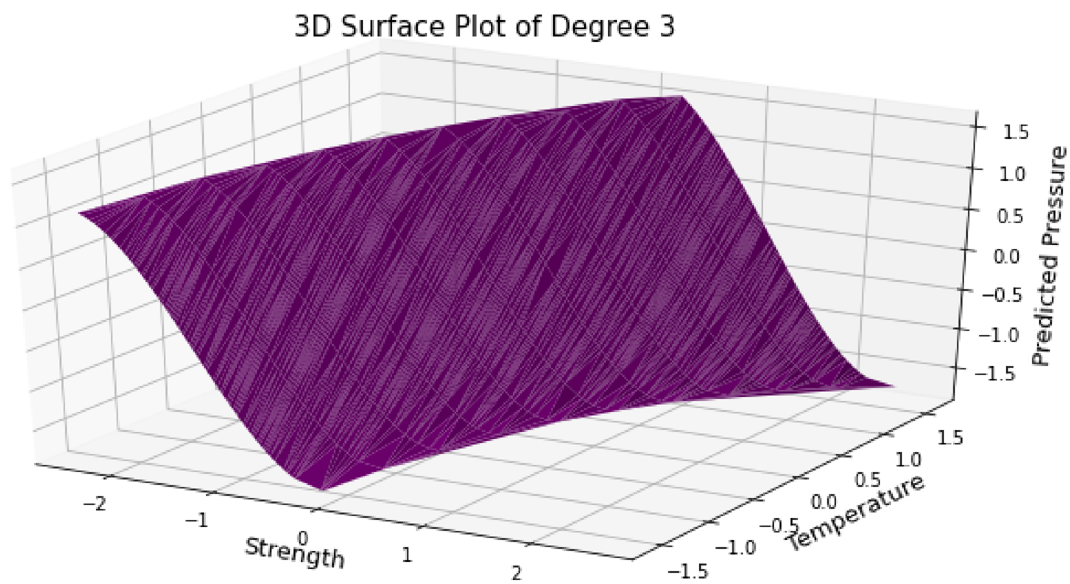
Using

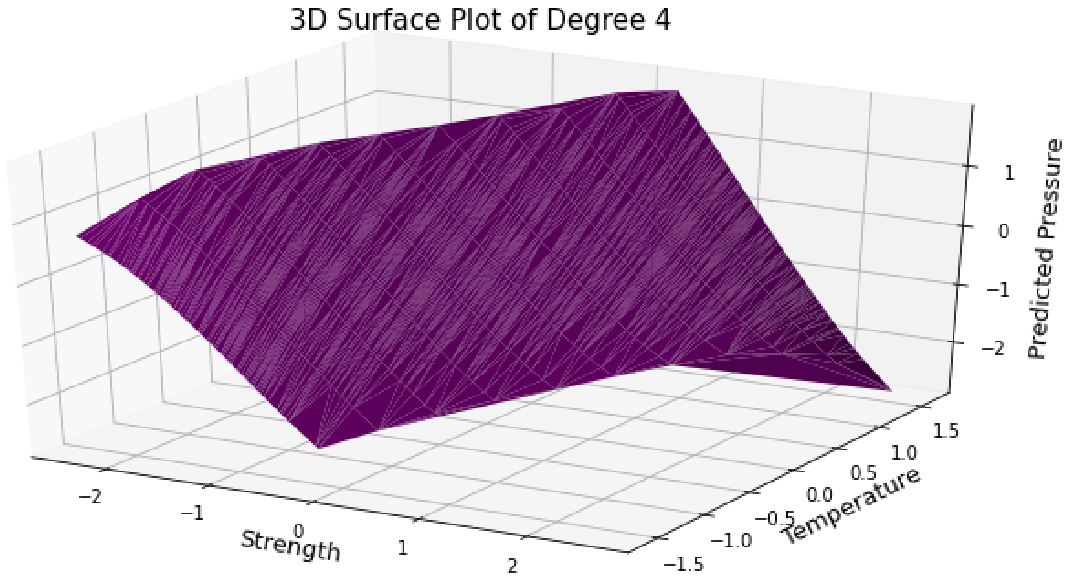
STOCHASTIC GRADIENT DESCENT ALGORITHM :

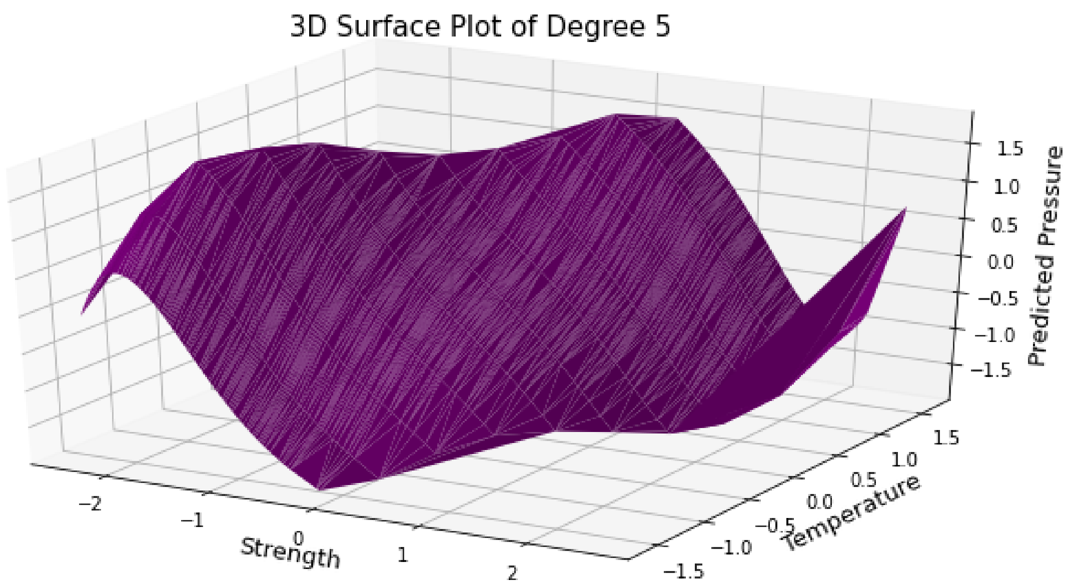


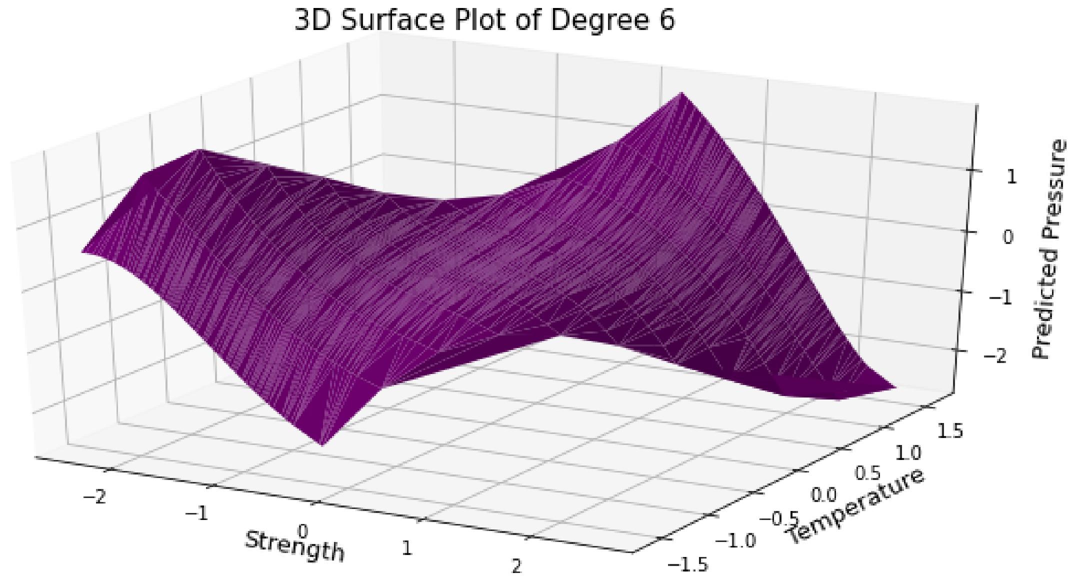


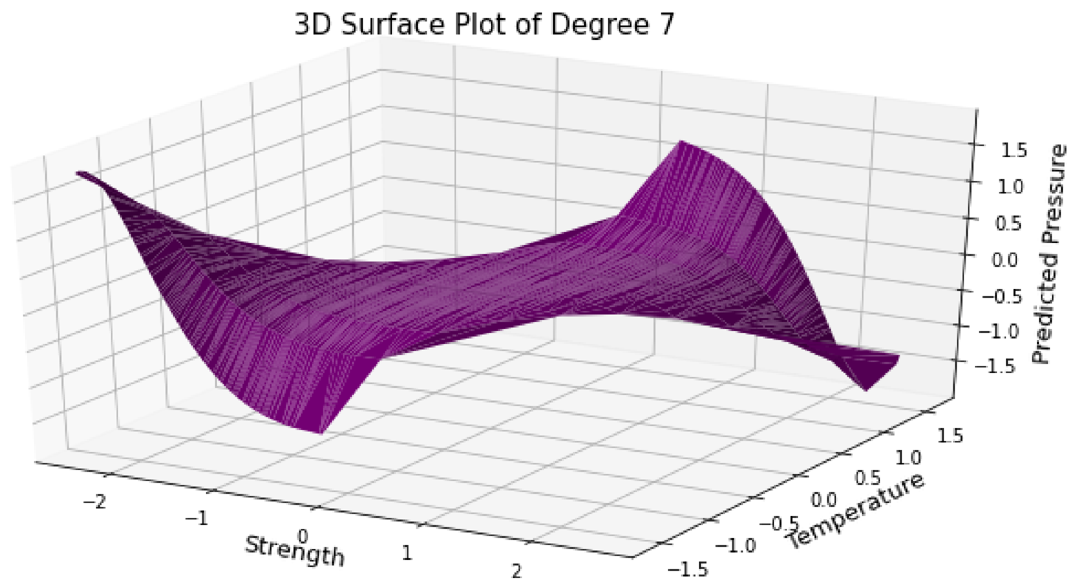

3D Surface Plot of Degree 2
Predicted Pressure
Strength
Temperature
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1
0
−1
−2
−3

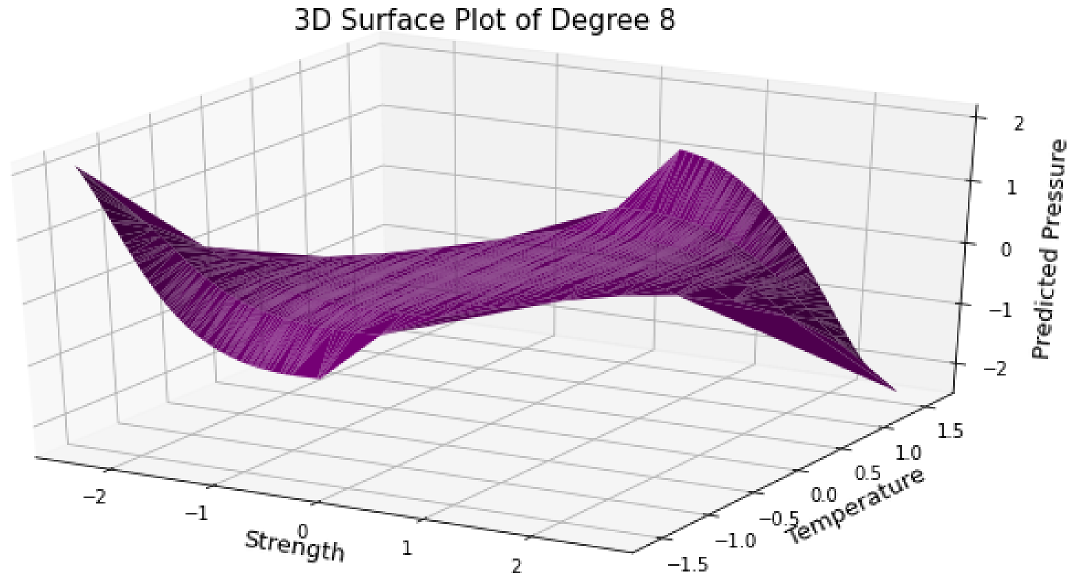

3D Surface Plot of Degree 3
Predicted Pressure
Strength
Temperature
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

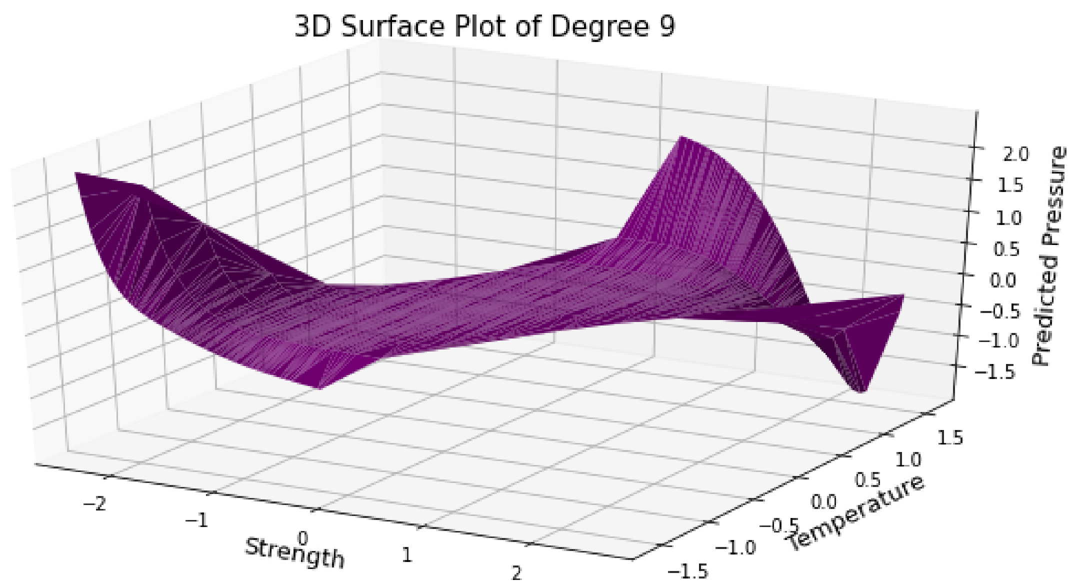

3D Surface Plot of Degree 4
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1
0
−1
−2


3D Surface Plot of Degree 5
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


3D Surface Plot of Degree 6
Predicted Pressure
1
0
−1
−2
Strength
−2
−1
0
1
2
Temperature
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


3D Surface Plot of Degree 7
Predicted Pressure
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Strength
−2
−1
0
1
2
Temperature
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


3D Surface Plot of Degree 8
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−2
−1
0
1
2


3D Surface Plot of Degree 9
Strength
Temperature
Predicted Pressure
−2
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

How does **overfitting** actually work?

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.

In overfitting TRAINING ERROR = 0

To overcome overfitting we use two methods :
Method 1 : Increase the data size (**training data**)
Method 2 : Constrain weights(W's) --- **regularization**

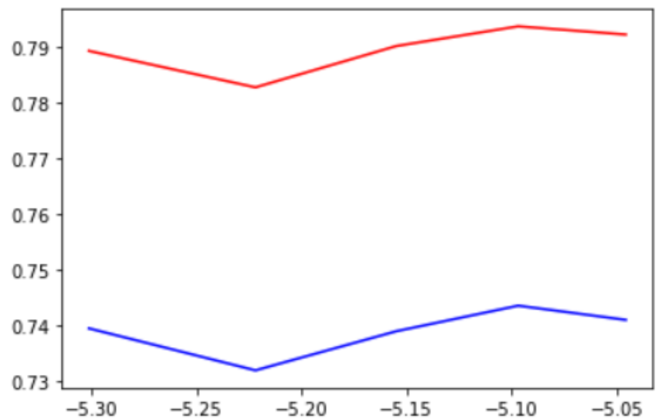
PART B :

Implementing **REGULARIZATION** for polynomial regression of degree 9

USING GRADIENT DESCENT ALGORITHM :

RIDGE REGRESSION:

Plot of the root-mean square error vs the logarithm of lambda

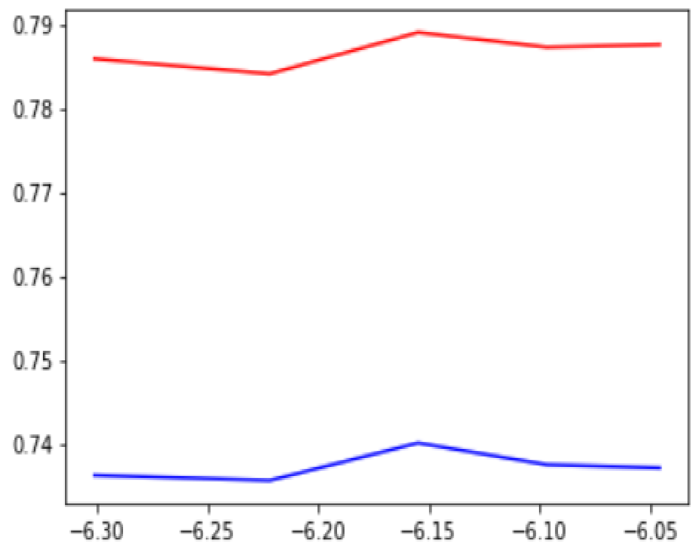


Minimum training and testing error

	0	1	2
0	0.000005	0.739375	0.789162
1	0.000006	0.731842	0.782620
2	0.000007	0.738860	0.790031
3	0.000008	0.743448	0.793568
4	0.000009	0.740900	0.792100

LASSO REGRESSION:

Plot of the root-mean square error vs the logarithm of lambda

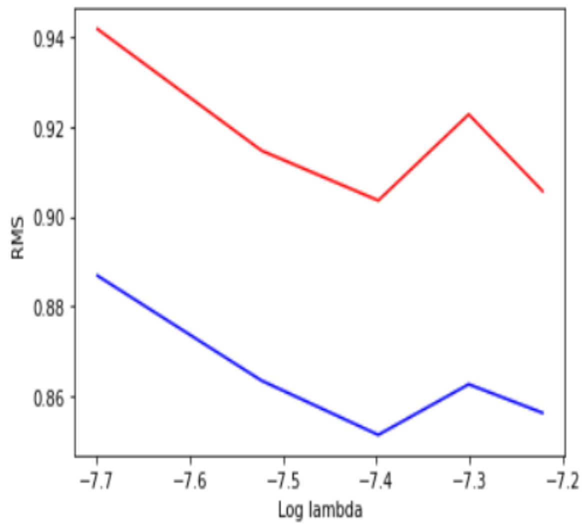


Minimum training and testing error

	0	1	2
0	1.000000e-07	0.736195	0.785856
1	4.000000e-07	0.735569	0.784068
2	7.000000e-07	0.740047	0.789000
3	1.000000e-06	0.737455	0.787241
4	1.300000e-06	0.737077	0.787546

<u>USING STOCHASTIC GRADIENT DESCENT ALGORITHM :</u>
<u>RIDGE REGRESSION:</u>

Plot of the root-mean square error vs the logarithm of lambda

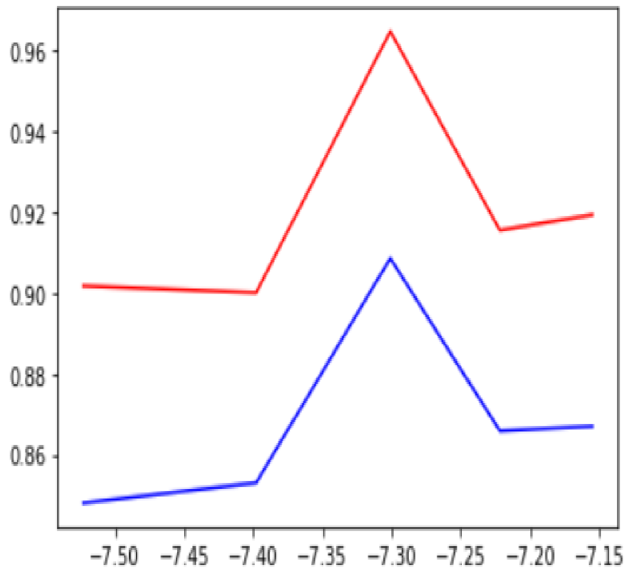


Minimum training and testing error

	0	1	2
0	2.000000e-08	0.886898	0.941726
1	3.000000e-08	0.863544	0.914671
2	4.000000e-08	0.851491	0.903594
3	5.000000e-08	0.862740	0.922739
4	6.000000e-08	0.856379	0.905674

LASSO REGRESSION:

Plot of the root-mean square error vs the logarithm of lambda



Minimum training and testing error

	0	1	2
0	3.000000e-08	0.847998	0.901738
1	4.000000e-08	0.852980	0.900087
2	5.000000e-08	0.908585	0.964786
3	6.000000e-08	0.865864	0.915546
4	7.000000e-08	0.866986	0.919377

The best model obtained in part a is Gradient descent with polynomial degree 3. The best model in part b is Ridge Regression with Gradient Descent with lambda being 0.000006.