FODS ASSIGNMENT 1

**RUTHVIK REDDY 2018B4A70701H**

**KESHAV CHANDA 2018B1A70757H**

**DEEPANSHU MODY 2018B1A70949H**

**Problem A:**

Gradient descent is used to train Linear Regression Model and this find model parameter needed. The algorithm is an iterative optimization algorithm capable of tweaking the model parameters by minimizing the cost function over the training data.

Stochastic Gradient Descent algorithm computes gradient using a single Training sample. As the name suggests it is stochastic in nature. The data sample should be in a random order, and this is why we want to shuffle the training set for every epoch.

We have first normalized our dataset and later split them into 70:30 ratio of training and testing data.

We then implemented the gradient descent algorithm and Stochastic Gradient Descent for the given dataset, and the degree of polynomials taken was from 0 to 9. We have found the Training error and Testing error for all the above polynomial. We have also plotted 3d surface plots for GD and SGD for all the 10 polynomials.

No regularization has been done as asked.

**The graphs and the values are reported below:**

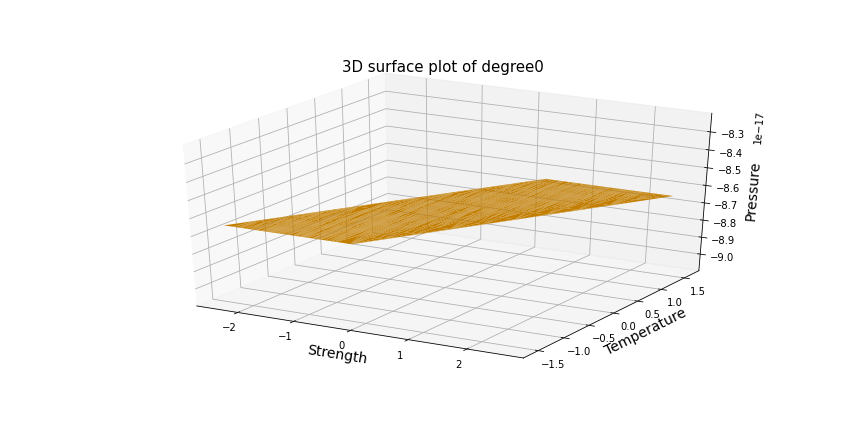
**Gradient Descent Algorithm**

Degree=0

Learning rate=0.5

Training Error=0.9991341991342051

Testing Error=0.9979797979798003

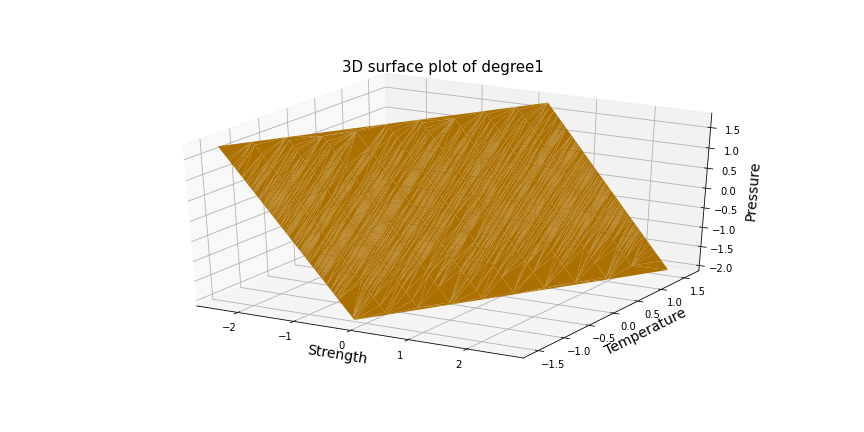


Degree=1

Learning rate=0.5

Training Error=0.20039024433663222

Testing Error=0.2009993643058269

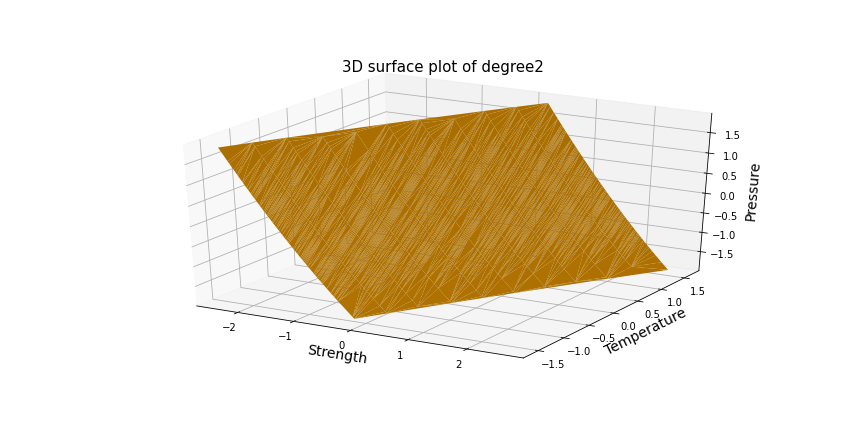


Degree2

Learning rate=0.1

Training Error=0.1988122016082634

Testing Error=0.1987885795221583

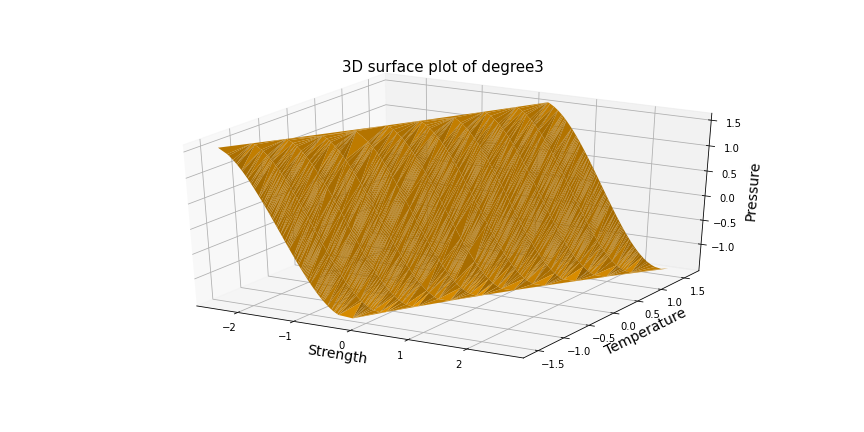


**Degree=3**

**Learning rate=0.04**

**Training Error=0.1854114390281051**

**Testing Error=0.1835889243390667**

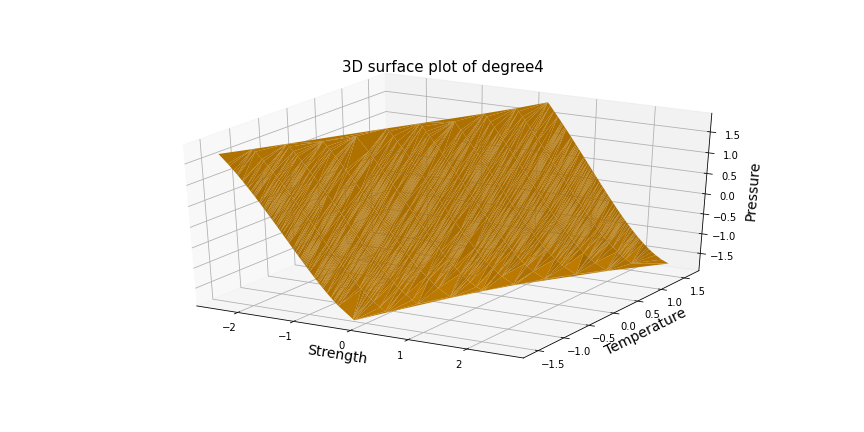


Degree=4

Learning rate=0.01

Training Error=0.1923512740775122

Testing Error=0.19243958153813914

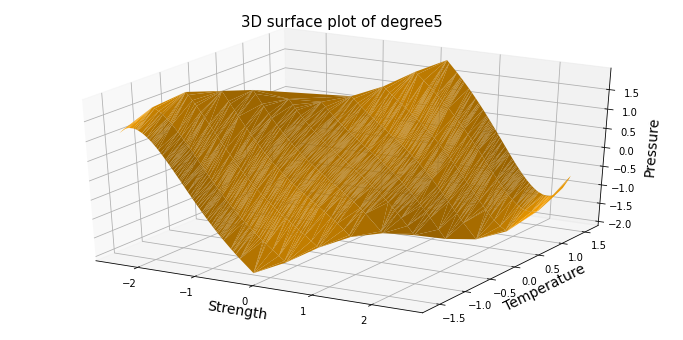


Degree=5

Learning rate=0.001

Training Error=0.22153178037139384

Testing Error=0.23366665605575068

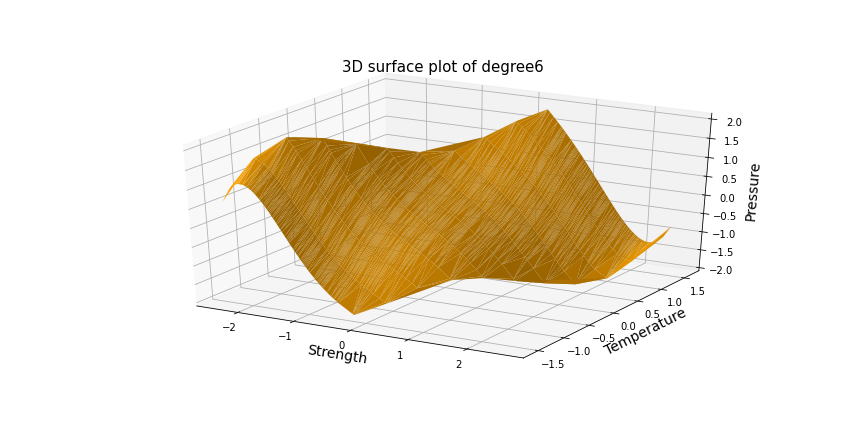


Degree=6

Learning rate=0.0008

Training Error=0.23343284158772468

Testing Error=0.22913134372466

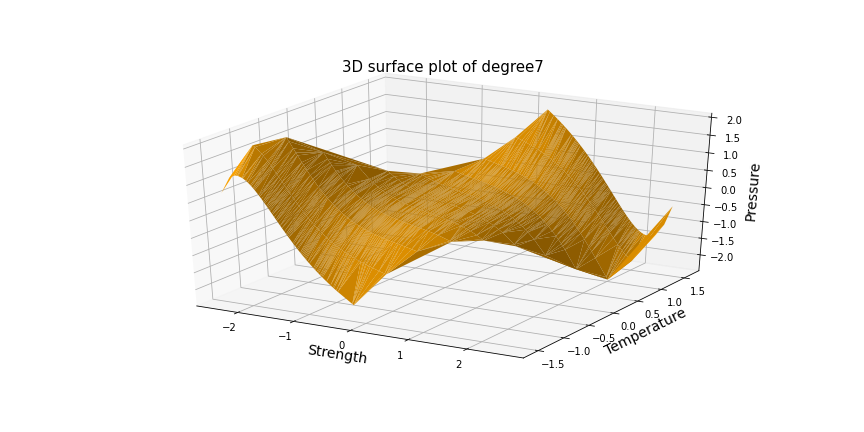


Degree=7

Learning rate=0.0015

Training Error=0.39204077967564055

Testing Error=0.38114083030159135

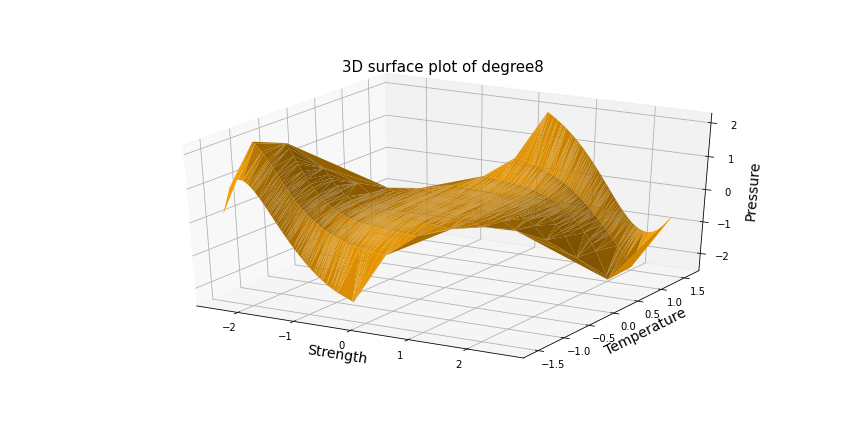


Degree=8

Learning rate=0.0003

Training Error=0.5465044360704969

Testing Error=0.5357583208013709

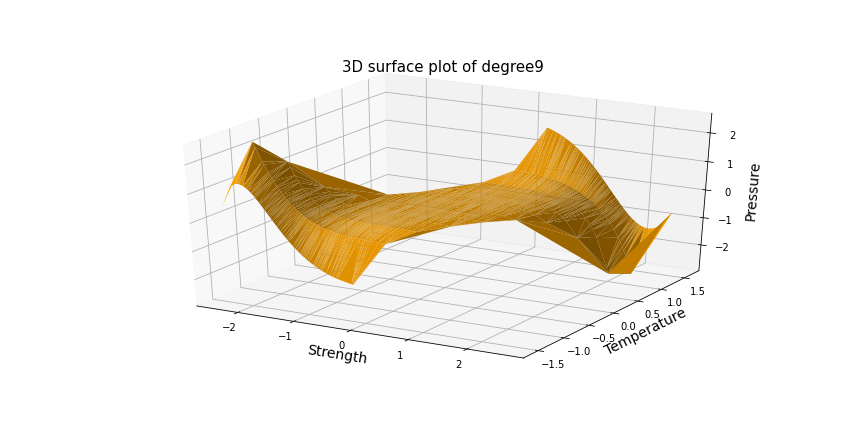


Degree=9

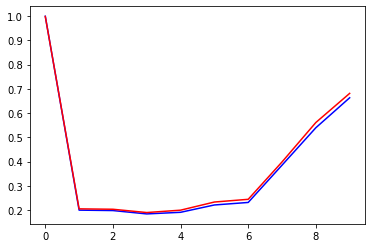
Learning rate=0.00004

Training Error=0.6621547852955012

Testing Error=0.6691033544339557



The following is the Error vs Degree graph for gradient descent. From the graph, we can conclude that degree=3 gives us the minimum error.



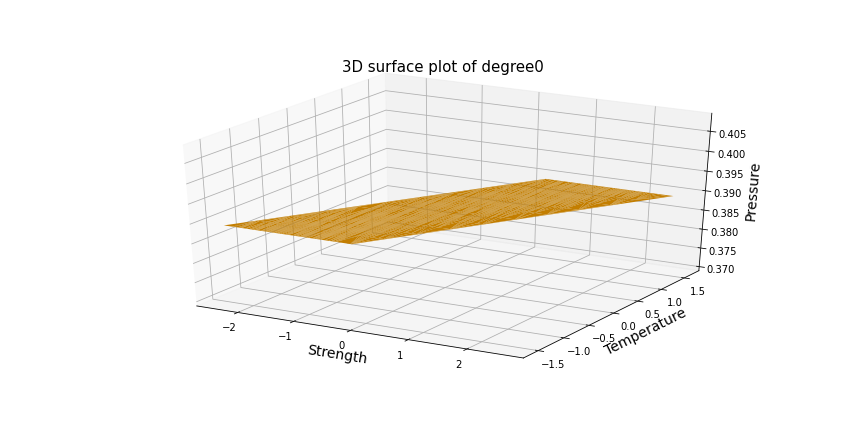
**Stochastic Gradient Descent Algorithm**

Degree=0

Learning rate=0.1

Training Error=1.0397321472828454

Testing Error=1.0385777461284404

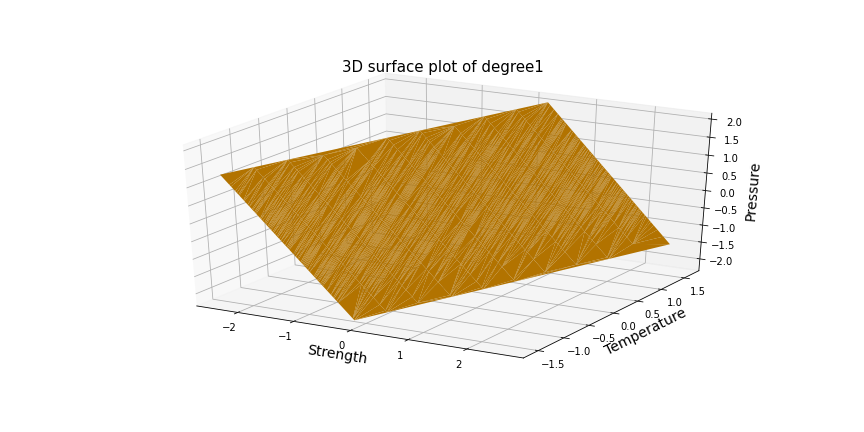


Degree=1

Learning rate=0.1

Training Error=0.22363514551109898

Testing Error=0.21582743726669523

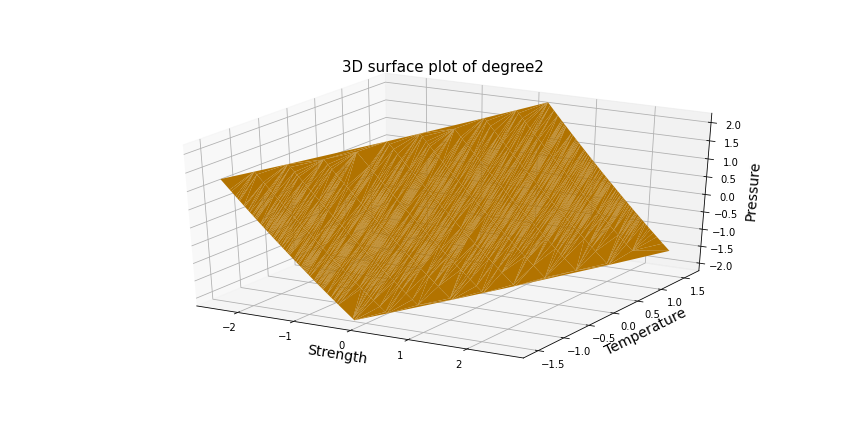


Degree=2

Learning rate=0.05

Training Error=0.2346001415385539

Testing Error=0.22293109212133036

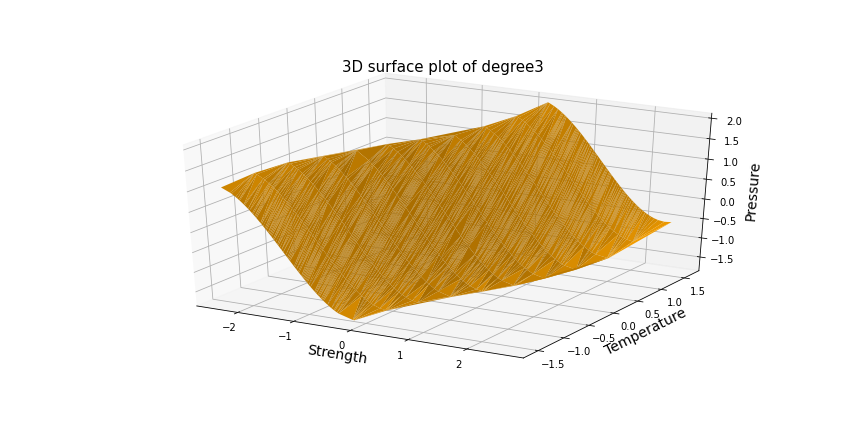


Degree=3

Learning rate=0.009

Training Error=0.20443375890825385

Testing Error=0.199086851483742

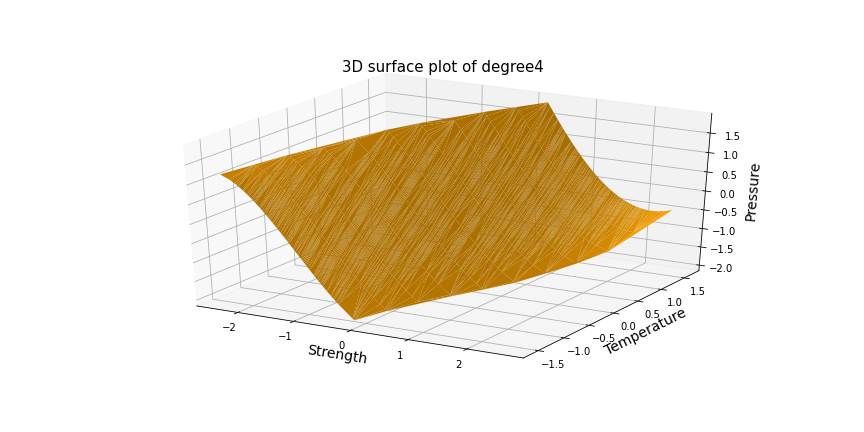


Degree=4

Learning rate=0.002

Training Error=0.21493321582208136

Testing Error=0.21252455398619965

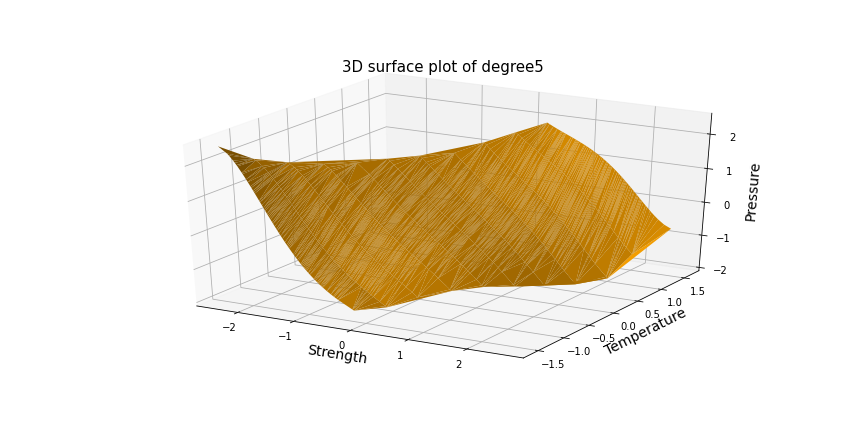


Degree=5

Learning rate=0.0005

Training Error=0.24536789910877838

Testing Error=0.24783876021015067

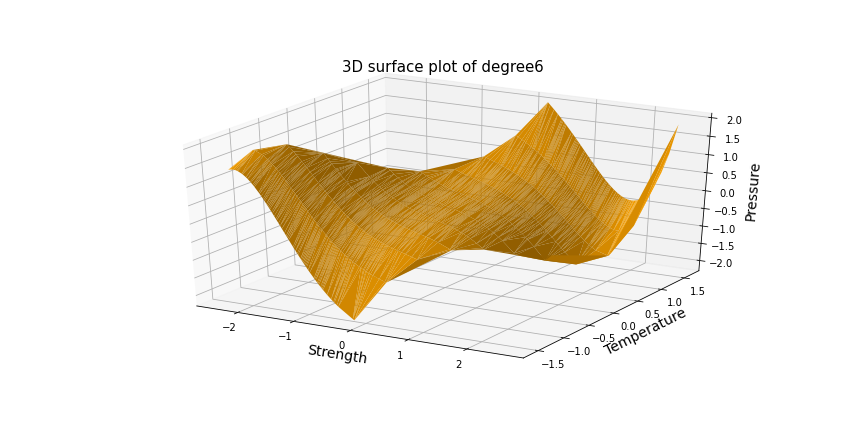


Degree=6

Learning rate=0.00007

Training Error=0.4054127040974734

Testing Error=0.3819965482491879

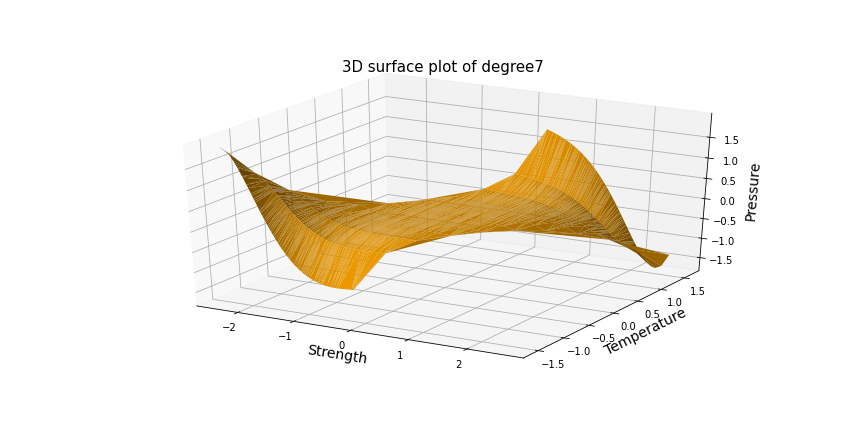


Degree=7

Learning rate=0.000001

Training Error=0.6856299669982977

Testing Error=0.6713481851096177

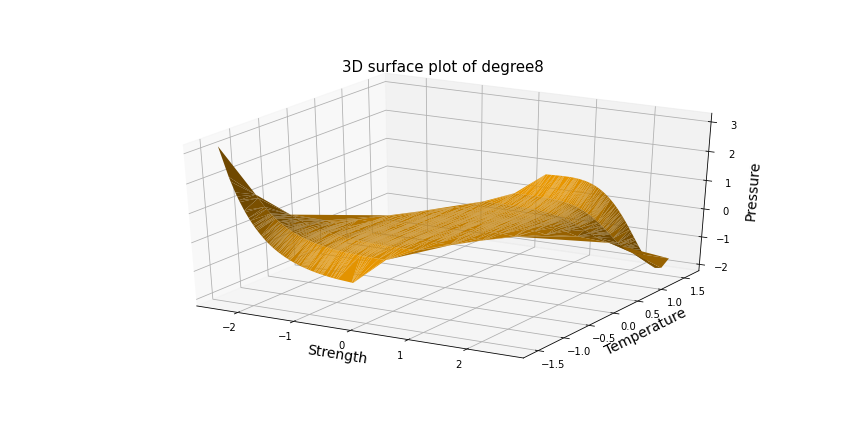


Degree=8

Learning rate=0.0000005

Training Error=0.7718862618992636

Testing Error=0.7812900260817236

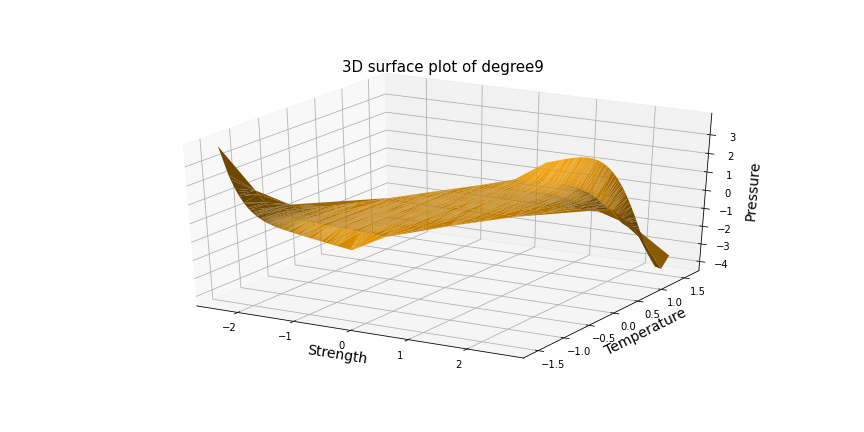


Degree=9

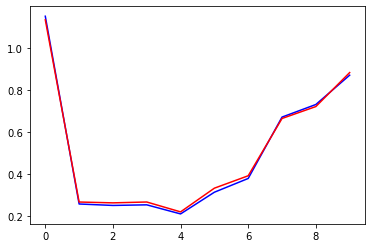
Learning rate=0.00000005

Training Error=0.7929197800732468

Testing Error=0.807596899612641



The following is the Error vs Degree graph for stochastic gradient descent. From the graph, we can conclude that degree=4 gives us the minimum error.



From the above graphs and the error values, we observe that the training and testing error is least for a **degree 3 polynomial**( It doesn’t overfit or underfit). Hence we think that a degree 3 polynomial is the best model. As the polynomial degree increases, the curve tries to align itself too much along with the training data, due to which it overly fits with respect to training data. This leads to an increase in testing errors.

**Problem B**We ran the regression model for 5 different values of lamda. Out of those, the optimal model has been figured out using the following plot.

**Training Error for 5 values of lambda respectively:** [0.5504683336160868, 0.5528305189766408, 0.5600924255864841, 0.556270004457822, 0.556039480203206]

**Testing Errors for 5 values of lambda respectively:** [0.53359087701208, 0.5359363612469779, 0.5437538439877767, 0.5406347684304531, 0.5407948052423562]

