**CS F320 FODS**

**Assignment 1**

**BY**

|  |  |
| --- | --- |
|  |  |
| **Chinni Vamshi Krushna** | **2021A7PS2084H** |
| **Atharva Chikhale** | **2021A7PS2752H** |

**Table of contents**

|  |  |
| --- | --- |
| **Content** | **Page No.** |
| Introduction……………………………………………………………………………… | 3 |
| Part A………………………….……………………………………………………………. | 4 |
| Task 1………………………….…………………………………………… | 4 |
| Task 2……………………………….……………………………………… | 5 |
| Task 3………………………………….…………………………………… | 6 |
| Task 4……………………………………………………….……………… | 8 |
| Part B………………………………………………………….………………………………. | 9 |
| Task 1…………………………………………………………….………… | 9 |
| Task 2………………………………………………………………….…… | 11 |
| Task 3……………………………………………………………….……… | 17 |
| Task 4…………………………………………………………….………… | 20 |

INTRODUCTION

This report contains analysis of the work done for Assignment 1 of CS F320 FODS course. The objective of this assignment is to develop predictive models for given dataset and use them to predict target variable using the feature variables. In Part A, we use polynomial regression to capture non-linear relationships in the data, and batch gradient descent as the optimization method and choose the best-fitting one.

In Part B, we conduct a comparative analysis of polynomial regression models to predict the target variable based on two input feature variables. The analysis includes the development of nine polynomial regression models with degrees ranging from 0 to 9, as well as choosing the best-fitting model and exploration of regularized polynomial regression models with different regularization parameters (q = 0.5, 1, 2, 4) and λ values for the best-fitting model. Both Stochastic and Batch Gradient Descent methods are applied to build and evaluate these models.

Part A

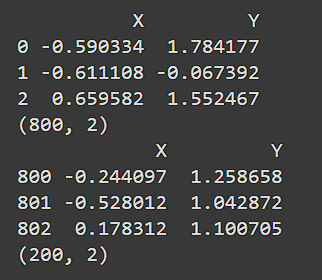
# Task 1: Data Preprocessing

The shared dataset was as following:



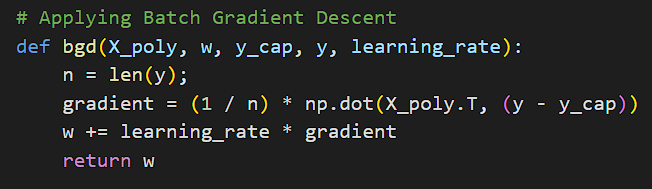
The dataset comprises 1,000 records with feature X and target variable Y. Data preprocessing included normalizing this data.

After processing (normalizing) the data using the formula: X’ = (X - µ) / σ, we shuffle and split it into 80:20 ratio and get the two datasets named `training` and `testing` as shown below:



# Task 2: Polynomial Regression

We constructed polynomial regression models with degrees ranging from 1 to 9. Each model was trained using the normal equation method to fit the data. We applied batch gradient descent with a learning rate of 0.001.



Below are the final training and testing LMS errors obtained:

Training:

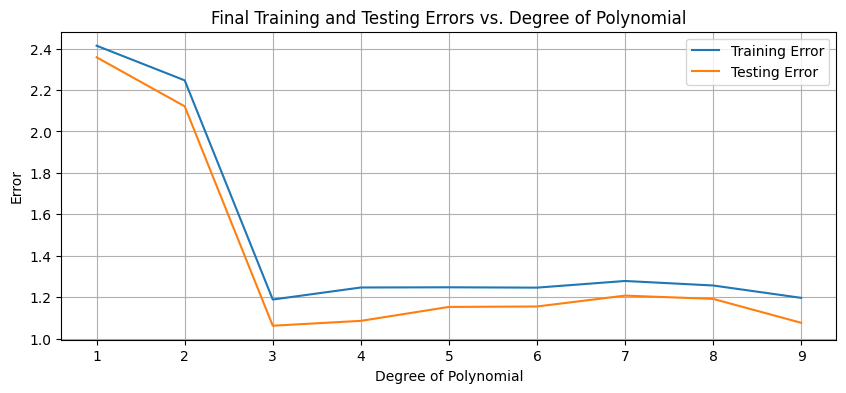
****

Testing:

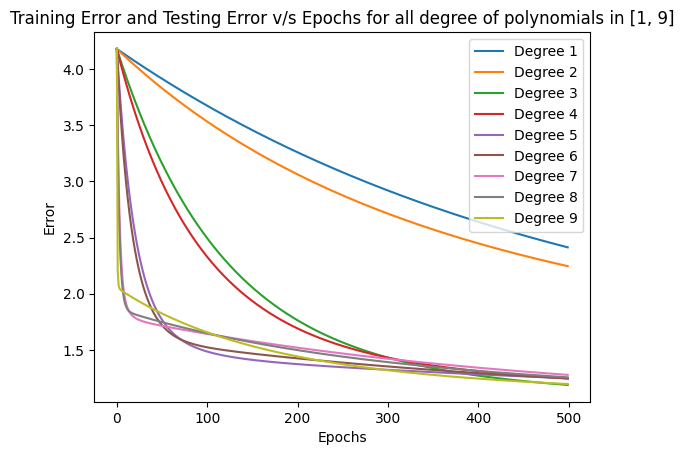


# Task 3: Graph Plotting

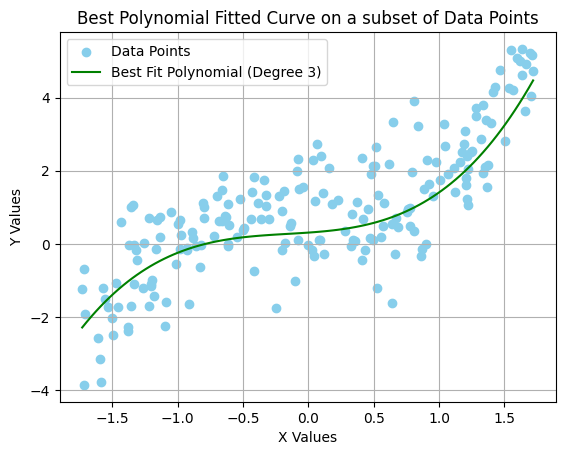
### Plot 1



### Plot 2



### Plot 3



# Task 4: Comparative Analysis

In this section, we perform a comparative analysis of the nine polynomial regression models based on the obtained training and testing errors and determine which model performs the best for our dataset.

**Training Error**: The training error generally decreases as the polynomial degree increases. Higher-degree models tend to fit the training data very closely.

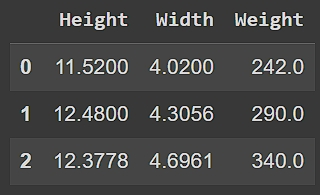
**Testing Error**: The testing error initially decreases from degree 1 to 3, indicating improved generalization. However, it starts to increase from degree 4 onwards, suggesting overfitting.

Considering the trade-off between training and testing errors, as well as the risk of overfitting, we selected the polynomial regression model with degree 3 for this dataset. It achieves a reasonably low testing error while avoiding the overfitting observed in higher-degree models.

Part B

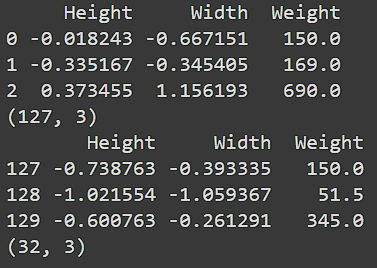
# Task 1: Data Preprocessing

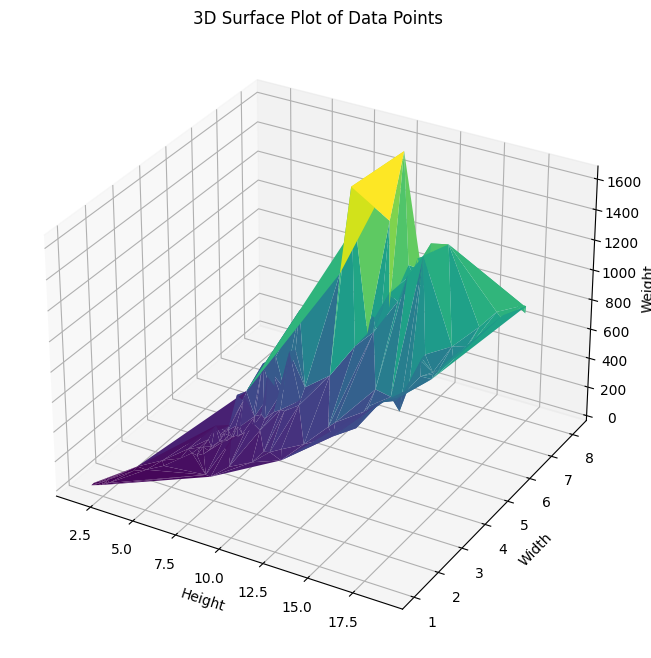
The shared dataset was as following:



The dataset comprises 169 records with features Height and Width of fish and target variable Weight of fish. Data preprocessing included normalizing this data and replacing null/Nan values with mean of the specific columns.

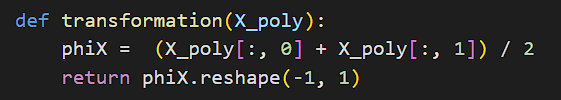
After processing (normalizing) the data using the formula: X’ = (X - µ) / σ and replacing null/Nan values, we shuffle and split it into 80:20 ratio and get the two datasets named `training` and `testing` as shown below:





# Task 2: Polynomial Regression

Nine polynomial models were developed. Feature variables were combined in Φ(X) by averaging them out. Below is the image of code snippet:



Batch gradient descent and LMS error functions remain same as for Part A.

Upon developing the model with learning rate = 0.00001 and 500000 iterations, using batch gradient descent method, following errors were obtained for training set and testing set:

Training:

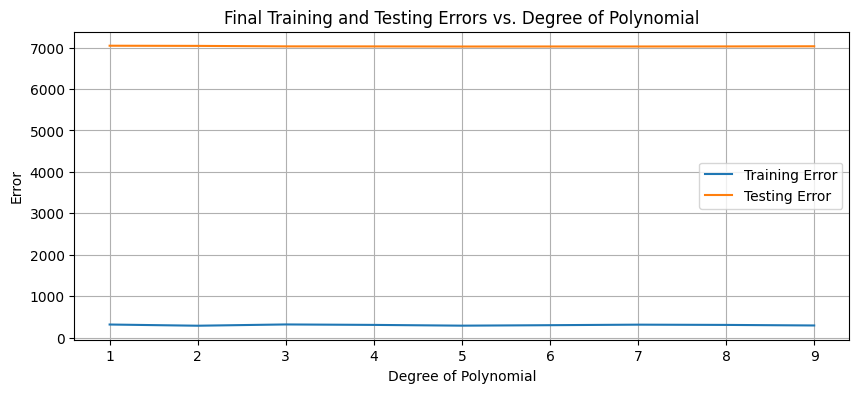


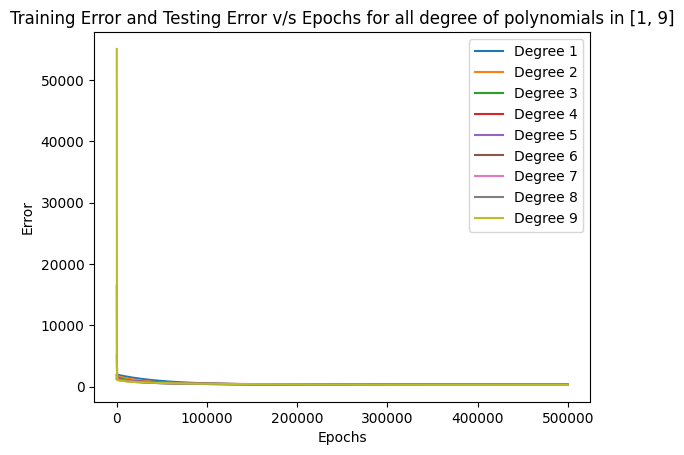
Testing:



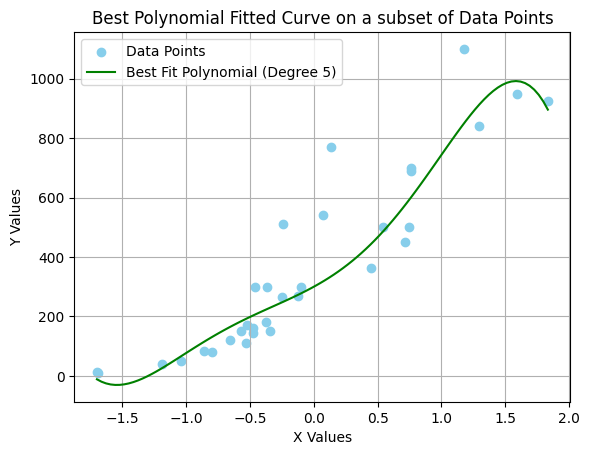
(From left to right, degree 1 to 9)

Using the results obtained above, following graphs were plotted to determine the best-fitting polynomial:

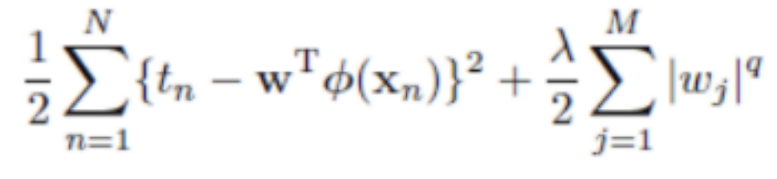




Upon examining the above plots, we observer that the errors remain almost constant for training and testing sets irrespective of the degree of polynomial. Upon closer look, the degree 5 has least error for the testing set and hence is the best-fitting curve. Plot below is a visualization of the same.

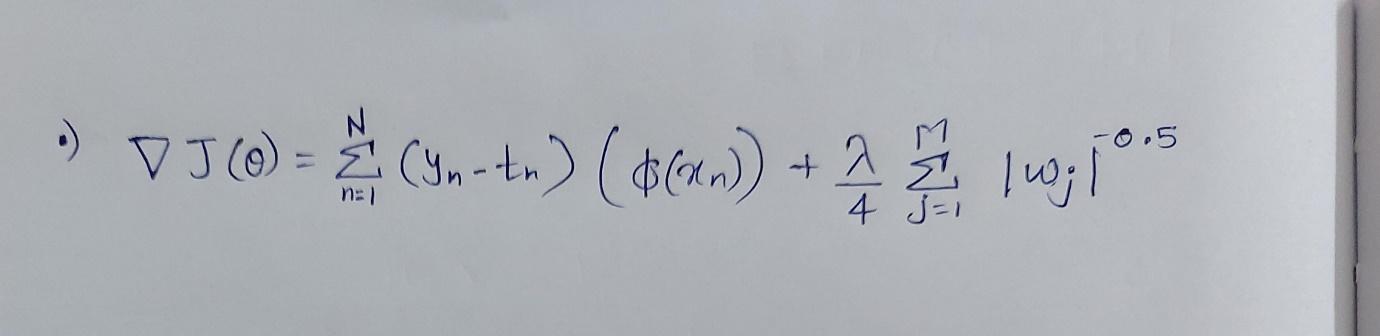


We proceed to build regularized polynomial regression models of degree 3 with different regularization parameters (q = 0.5, 1, 2, 4) for both Stochastic and Batch Gradient Descent using the given equation for errors:

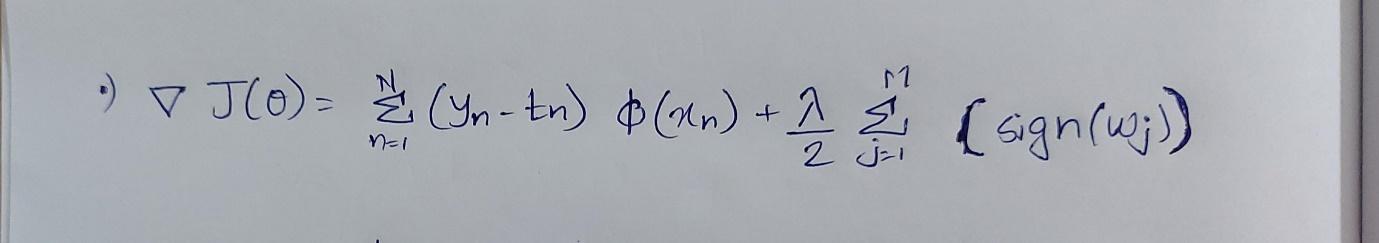


Gradients obtained after differentiating the equation after inserting the values of q:

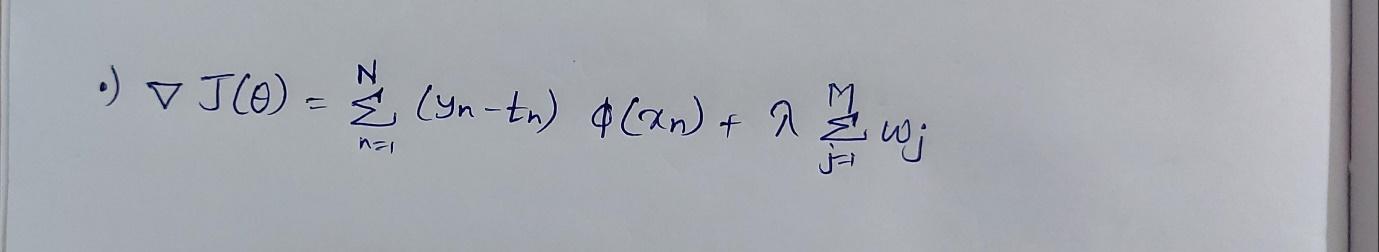
### Q = 0.5



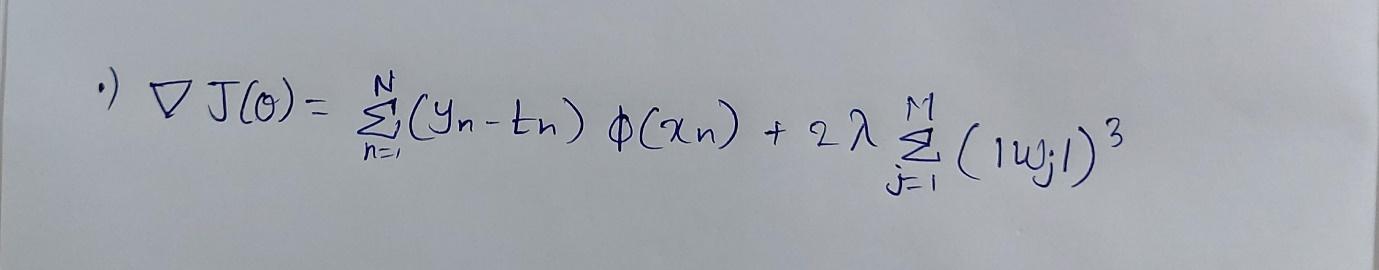
### Q = 1



### Q = 2



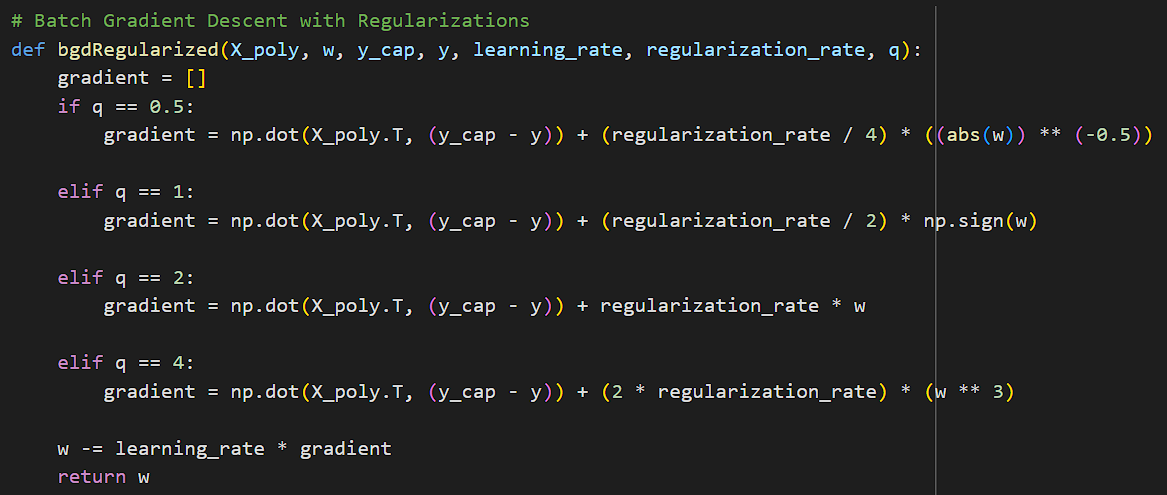
### Q = 4



To find the optimal models, we experimented with λ values in the range [0, 1] for each value of q. Cross-validation was employed to select the best λ for each model for both Batch and Stochastic Gradient Descent.

### Batch Gradient Descent

New BGD function was implemented as follows:



Training Errors:



Testing Errors:

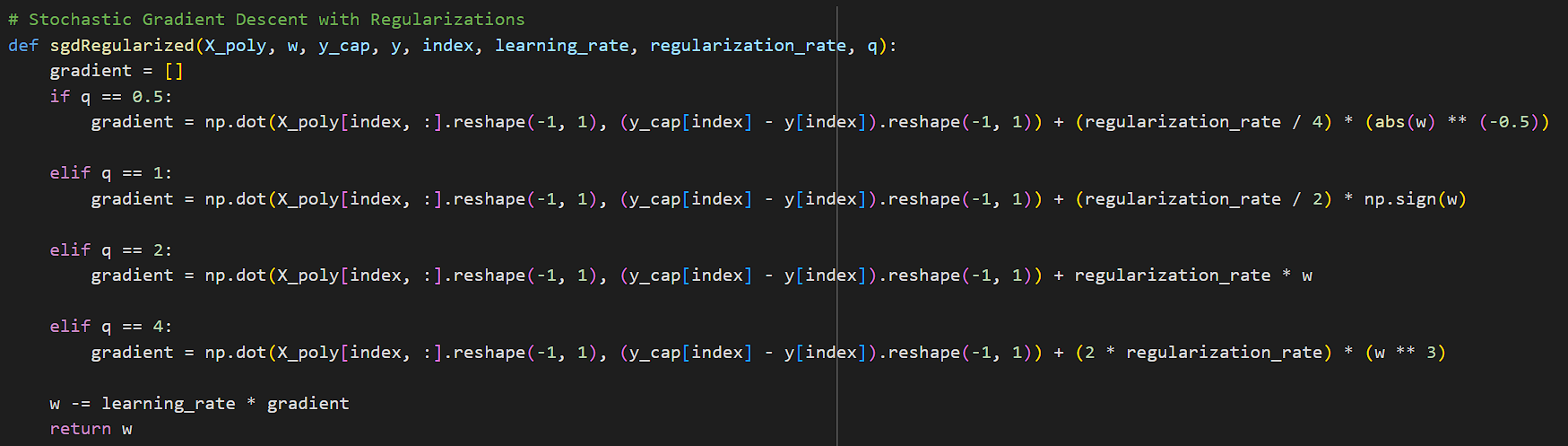


Learning rate and λ values for each Q in BGD:

|  |  |  |
| --- | --- | --- |
| **Q** | **Learning Rate** | **λ** |
| 0.5 | 0.0001 | 0.001 |
| 1 | 0.0001 | 0.001 |
| 2 | 0.0001 | 0.000001 |
| 4 | 0.0001 | 0.00000000001 |
|  |  |  |

### Stochastic Gradient Descent

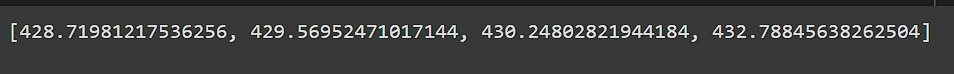
SGD for regularization was implemented as following:



Training Errors:



Testing Errors:



Learning rate and λ values for each Q in SGD:

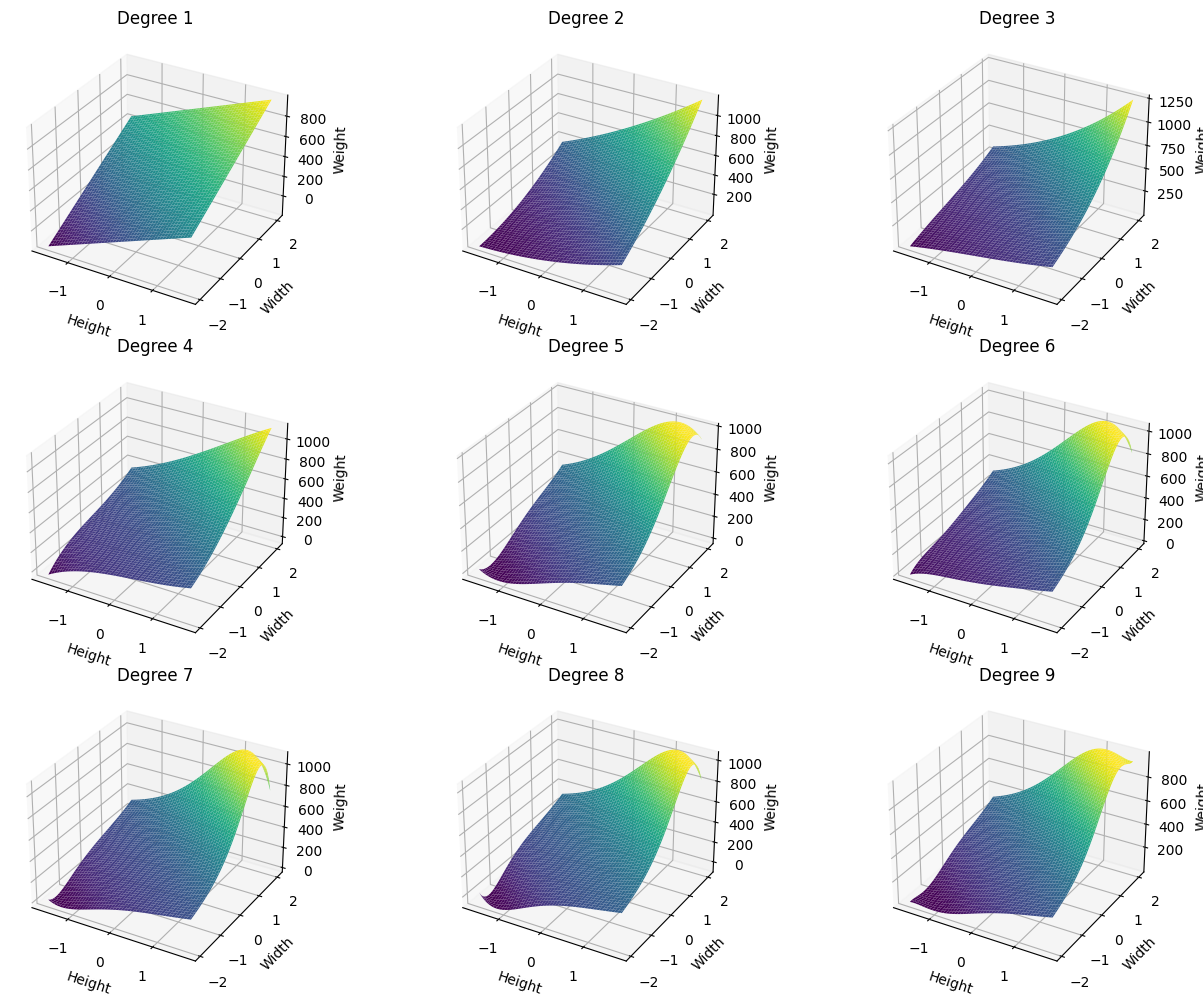
|  |  |  |
| --- | --- | --- |
| **Q** | **Learning Rate** | **λ** |
| 0.5 | 0.0001 | 0.000001 |
| 1 | 0.0001 | 0.0000001 |
| 2 | 0.0001 | 0.00000000001 |
| 4 | 0.0001 | 0.00000000000001 |

It was observed that for the same number of iterations, BGD gives less LMS error compared to SGD. However, upon increasing the iterations of SGD, we get the same errors for it.

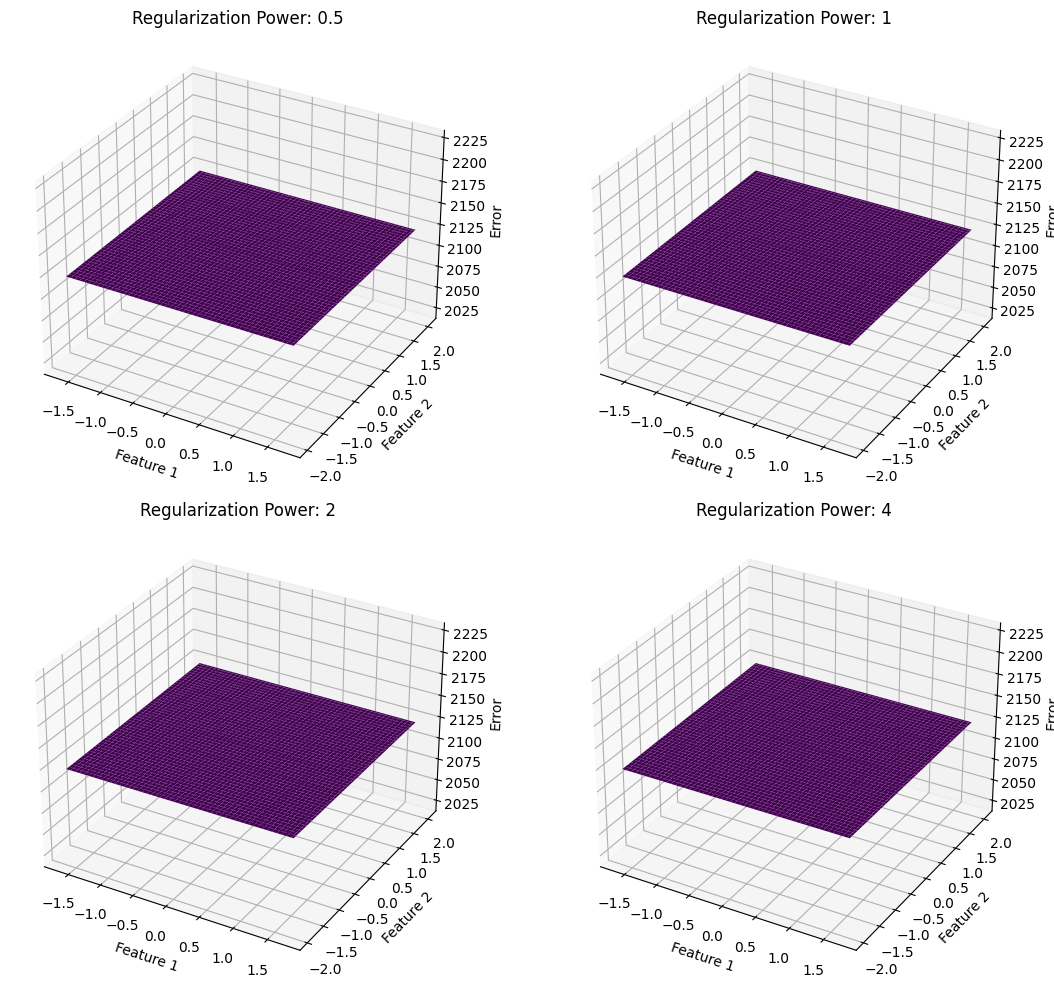
# Task 3: Graph Plotting

We generated surface plots for the nine polynomial regression models (degrees 0 to 9) and the four optimal regularized linear regression models using the Matplotlib library in Python.

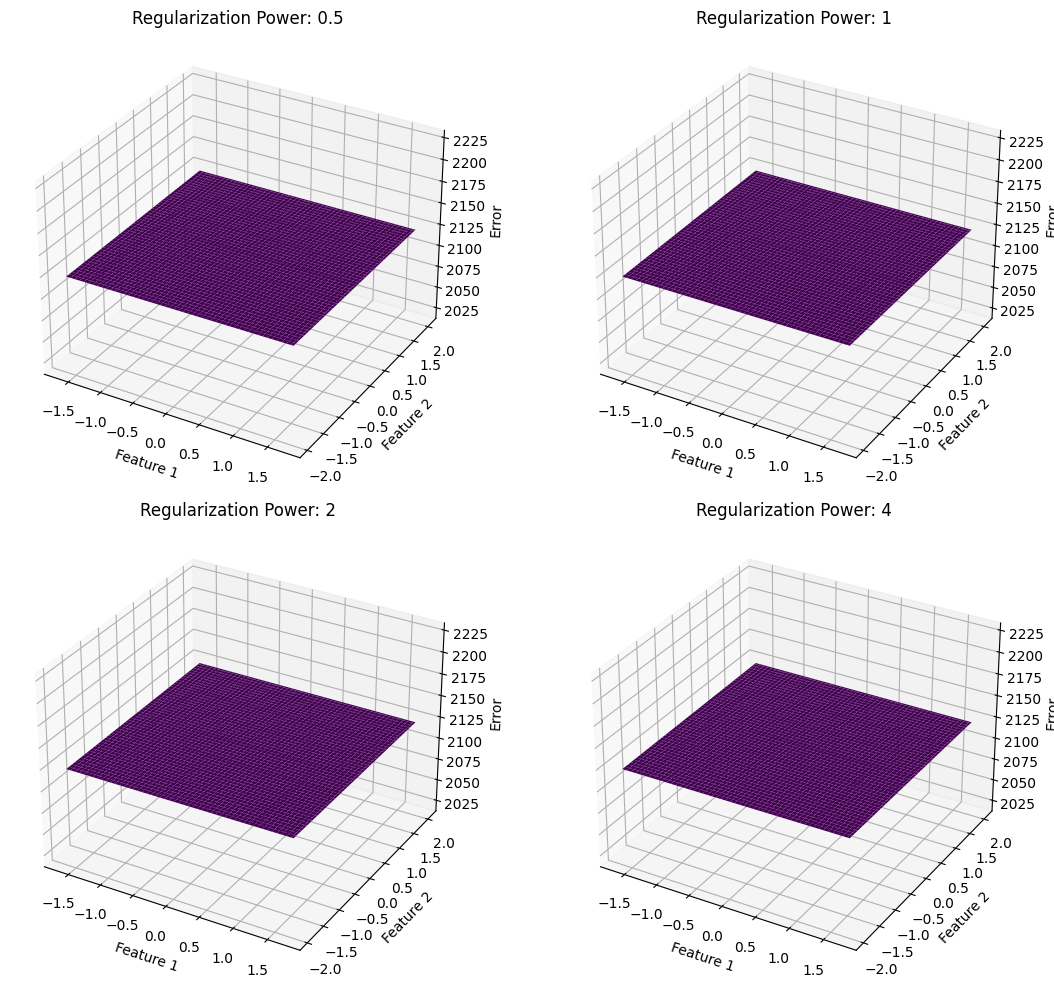
### Plot 1: Surface Plots for Non-Regularized Polynomials



### Plot 2: Surface Plots for Regularized Polynomials (Batch Gradient Descent)



### Plot 3: Surface Plots for Regularized Polynomials (Stochastic Gradient Descent)



# Task 4: Comparative Analysis

In this analysis, we developed nine polynomial regression models with degrees ranging from 0 to 9 to predict a target variable based on two input features. Next, we constructed regularized polynomial regression models with different generalized error functions (parameterized by q) and experimented with various λ values between 0 and 1 to find the optimal model for each value of q. We employed both Stochastic and Batch Gradient Descent methods and reported the best models obtained through these approaches for each value of q.

Here are the training and testing errors (MSE) for the SGD method with different q values, learning rates, and λ values:

q Learning Rate λ Training Error (MSE) Testing Error (MSE)

0.5 0.0001 0.000001 278.437 428.720

1 0.0001 0.0000001 278.485 429.570

2 0.0001 0.00000000001 278.523 430.248

4 0.0001 0.00000000000001 278.729 432.788

Here are the training and testing errors (MSE) for the BGD method with different q values, learning rates, and λ values:

q Learning Rate λ Training Error (MSE) Testing Error (MSE)

0.5 0.0001 0.001 271.268 444.481

1 0.0001 0.001 271.697 444.910

2 0.0001 0.000001 271.386 444.598

4 0.0001 0.00000000001 271.449 444.662

Observing the training and testing errors, we notice that as the value of q increases, given the sufficient values of λ, the training and testing errors remain almost the same. Upon further changing the values, the decrease in errors drops significantly and hence these values were taken as final errors for the respective q and gradient descent methods.

Conclusion:

.In the absence of regularization, the degree 5 polynomial model performs the best, but it suffers from overfitting when applied to unseen data. This is evident from the increase in testing errors beyond degree 5.

.The introduction of regularization helps mitigate overfitting by reducing training errors. However, the models still struggle to generalize well to unseen data, as evidenced by the relatively high testing errors.

.Regularization is a valuable tool for improving the robustness of polynomial regression models, but it may not entirely eliminate the overfitting problem. Model selection and hyperparameter tuning remain critical for achieving the best trade-off between fitting the training data and generalizing to new data.