Documentation for 1-B

CS F320 Foundations of Data Science

Assignment-1

Question: 1-B

Assignment 1-B Polynomial Regression and Regularization

Problem Statement

- The given dataset is a fish dataset consisting of ~200 data points, each having two feature variables and one continuous target variable.

Feature 1 : Width Of Fish **Feature 2 :** Height Of Fish **Target Feature :** Weight Of Fish

Group Members

Group Member	ID
Teerth Vasant Patel	2021A7PS2090H
Manthan Patel	2021A7PS2691H
Shrey C Paunwala	2021A7PS2808H

Documentation for 1-B

Question 1-B

Task 1: Data Preprocessing

- We Loaded the data file in a Panda DataFrame.
- Normalize the Feature Variables to ensure consistent scales across the dataset.
- **FORMULA USED** in Normalization: $X' = (X - \mu) / \sigma$
  Where μ is the mean and σ is the variance.
- To handle Missing value, they are predicted using the mean of the existing values in the corresponding feature. The function used in achieving the task is fillna() function.
- The dataset is shuffled to randomize the order of samples, preventing any order-related biases during training.

Task 2: Polynomial Regression

- We've constructed nine polynomial regression models, ranging from degree 0 to 9. These models make predictions about the target variable using two input features. The goal is to figure out which polynomial degree provides the best fit for our data.
- As stated in the problem statement, we've applied batch gradient descent.
- From the graph and table, it can be inferred that polynomial of degree 1 is the most suitable solution.

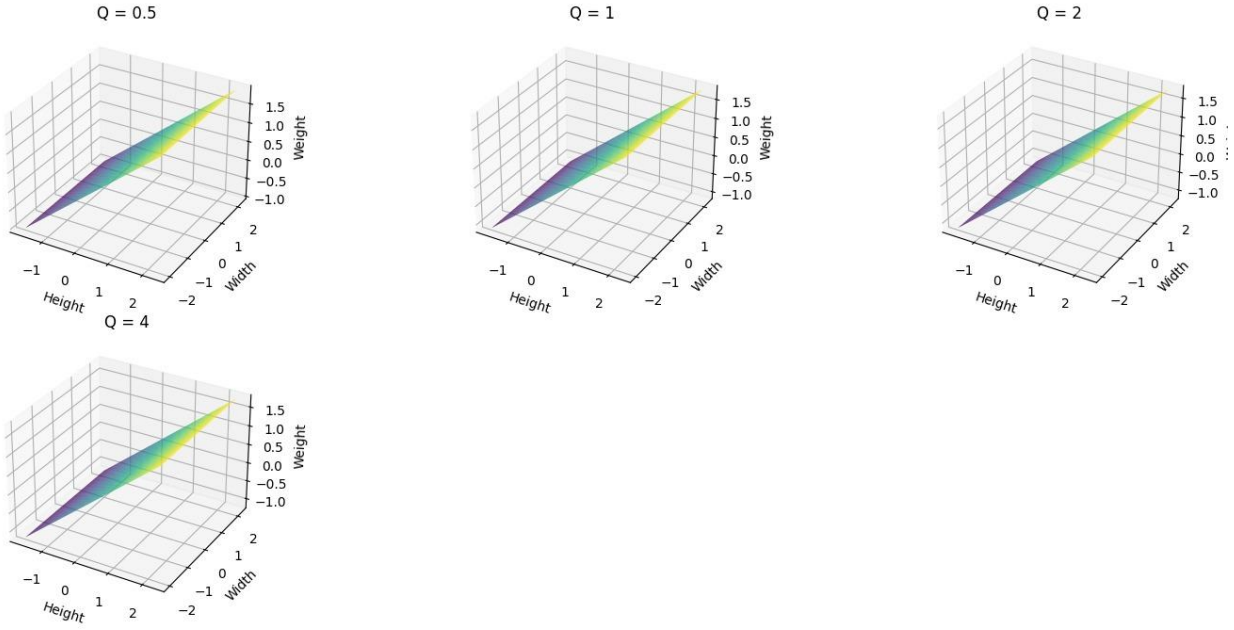
Documentation for 1-B

Degree	Training Error	Testing Error
1	0.8000031172641918	0.654925805104805
2	2.364849932206292	2.343161294978866
3	5.854475993415493	6.378649672374112
4	15.149790409698662	16.487209862889287
5	29.328665866667144	31.299846917517627
6	6.786107842668389	5.999890448560132
7	6.322493885112769	5.033436604603985
8	18.469627764739837	15.674326056625507
9	55.26135446206063	61.41434081854832

Documentation for 1-B

Batch Gradient Descent

Gradient Descent is an iterative optimization algorithm used for minimizing a function, often applied in machine learning to minimize a model's cost function. It calculates the gradient (derivative) of the function at a specific point and takes steps proportional to the negative gradient, iteratively moving towards the function's minimum. Various versions of Gradient Descent exist, including Batch Gradient Descent, Mini-batch Gradient Descent, and Stochastic Gradient Descent (SGD).

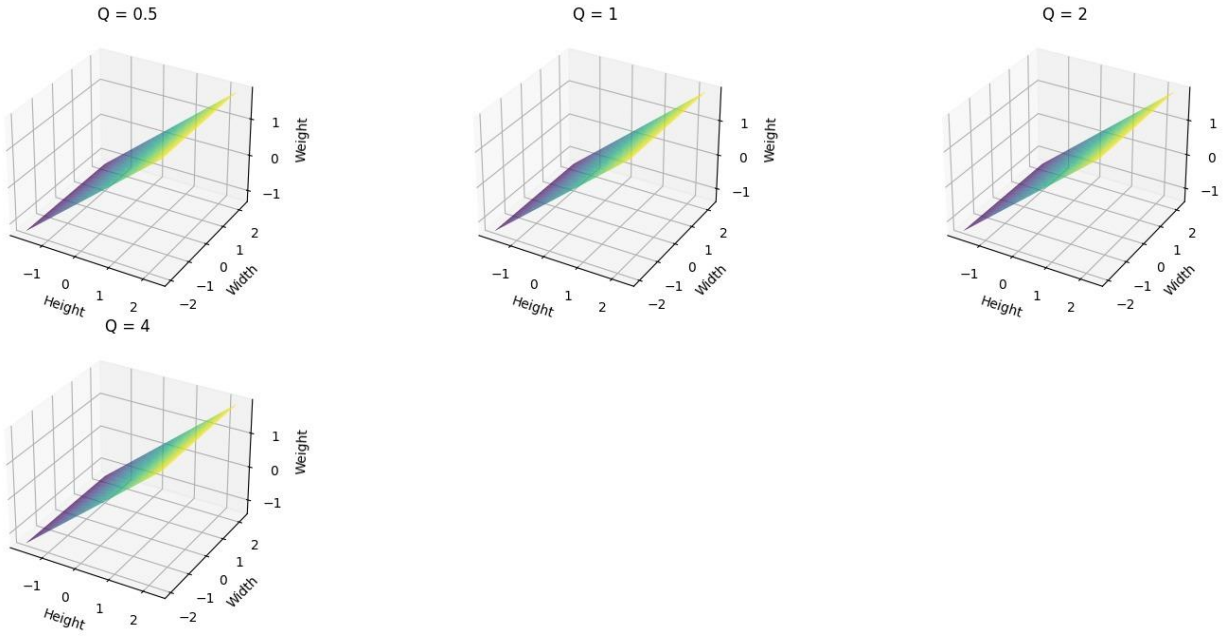


Best MSE: 0.2221849260314163
Best Q: 4

Documentation for 1-B

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a version of Gradient Descent, specifically designed for handling large datasets. Unlike Batch or Mini-batch Gradient Descent, which updates the model's parameters using batches of training examples, SGD updates the parameters one training example (or a small subset) at a time. This characteristic allows SGD to converge faster. However, because it updates parameters after each individual data point, the path to the minimum can be noisier. Despite this randomness, SGD is widely employed in machine learning due to its speed and its ability to navigate past local minima and saddle points.

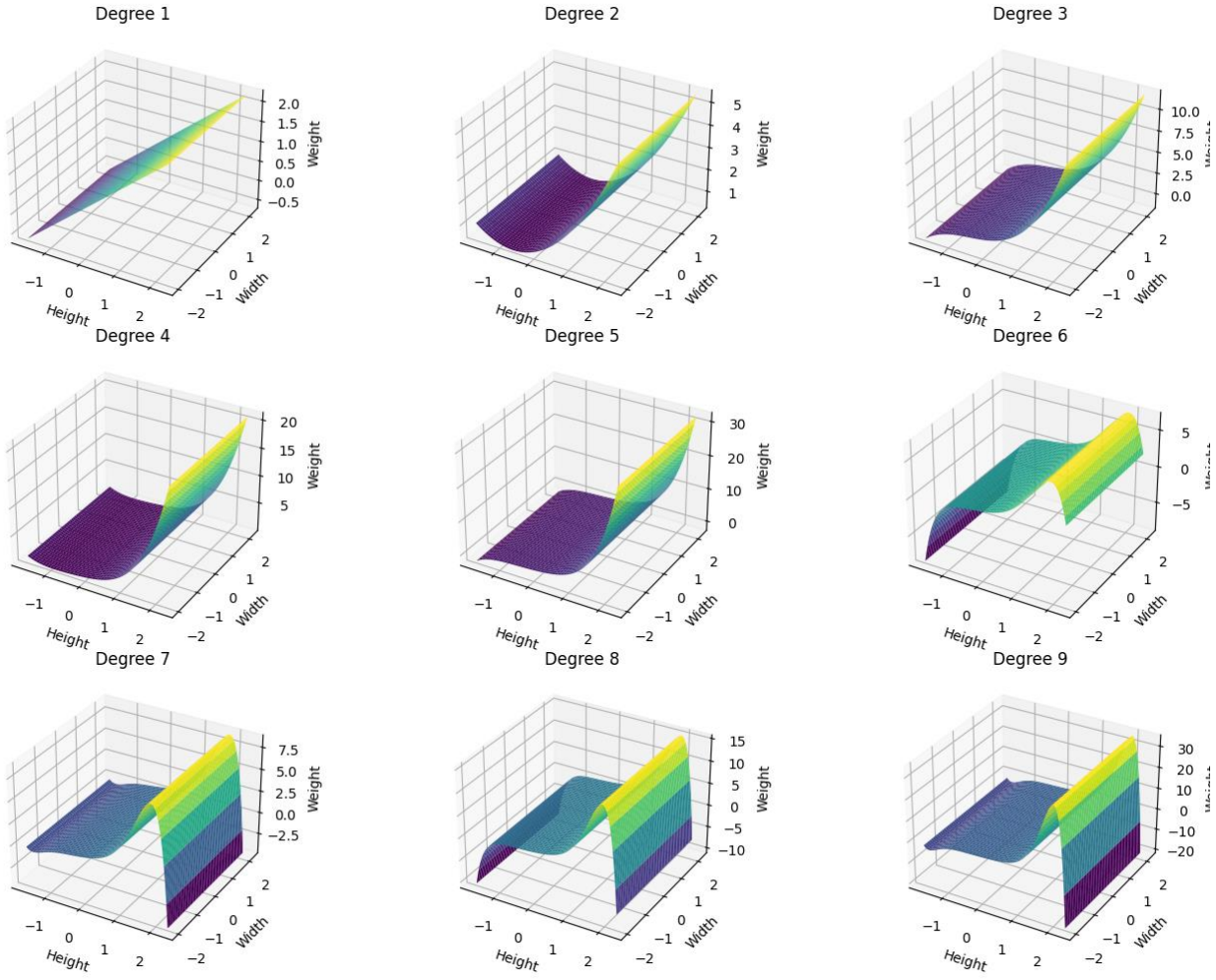


Best MSE: 0.23415358107583728
Best Q: 1

Documentation for 1-B

Task 3: Graph Plotting

Surface plots for all nine polynomial regression models are shown below.



Documentation for 1-B

Task 4: Comparative Analysis

A comparative analysis was carried out, exploring different values of lambda and powers to identify the best polynomial fit and its associated regularization factor. The results indicated that when lambda was set to 0.01, both training and testing errors were equal. This suggests that the introduction of regularization led to a reduction in testing errors but an increase in training errors.
The point where training and testing errors matched indicated the optimal lambda value. In our code snippet, it was observed that regardless of the power value used, setting lambda to 0.01 consistently resulted in both minimized training and testing errors being the same.

For Batch Gradient Descent:

Q(power)	**Training Error**	**Testing Error**
0.5	0.5724842843818322	0.31601356384260193
1	0.542527961360367	0.24590193214492867
2	0.5385596025507943	0.22769379942804363
4	0.5379998584362397	0.2221849260314163

For Stochastic Gradient Descent:

Q(power)	**Training Error**	**Testing Error**
0.5	0.18713619417522057	0.259001329001875
1	0.13788560390989546	0.23415358107583728
2	0.10129060993836186	0.24085837077899952
4	0.06635827836601861	0.26543219860429756