**Foundations of Data Science Assignment-1**

Assignment Report

Adit Rastogi - 2022A7PS1330H

Bhupathiraju Vishal Varma – 2022A7PS0174H

Jeeru Harshith Reddy – 2022A7PS0233H

Approach:

Task-1 Data-Preprocessing

In this task, we processed a dataset to prepare it for further analysis and model training. The main steps involved loading the data, normalizing the feature variable, splitting the data into training and test sets, and visualizing the data.

* **X**: The feature or independent variable.
* **Y**: The target or dependent variable.

To ensure randomness and eliminate any potential ordering biases in the dataset, we shuffled the dataset.

Also The dataset was divided into three sets:

* **Training set**: 600 rows (80% of the data)
* **Test set**: 200 rows (20% of the data)
* **Validation set**: 200 rows (used for further validation in future tasks)

The code used for this is as follows:

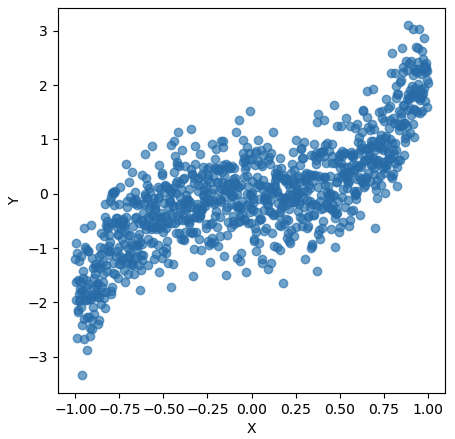


To normalize the feature variable X, the formula applied is

A black background with white text

Description automatically generated

Data Visualization



The scatter plot shows a general positive relationship between X and Y, suggesting that as X increases, Y also tends to increase. However, there is some spread and noise in the data, indicating that the relationship might not be perfectly linear.

In Task 1, we successfully prepared the data for further analysis by shuffling, normalizing, and splitting it into training, test, and validation sets. The scatter plot provided us with an initial understanding of the relationship between the feature variable **X** and the target variable **Y**, indicating a positive correlation, although with some noise

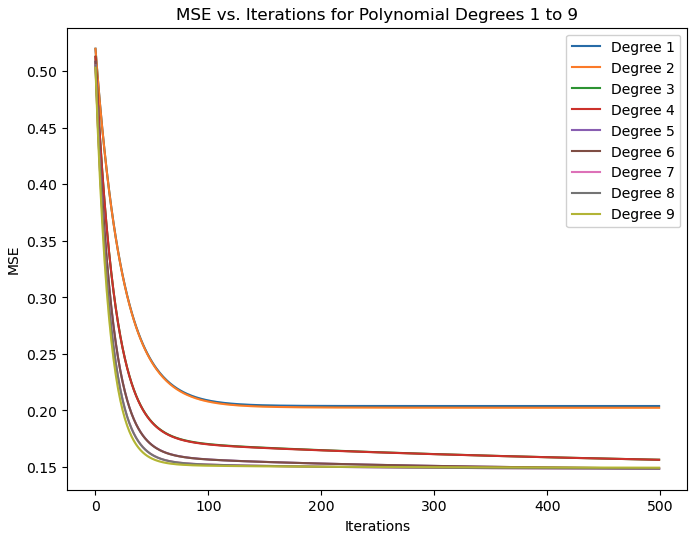
Task-2 Polynomial Regression and Regularization

This task explores the impact of polynomial regression models with varying degrees and L2 regularization (Ridge Regression) on a dataset. We aim to understand the behaviour of polynomial models from degree 1 to 9 and analyse how adding regularization helps manage the bias-variance trade-off for each polynomial degree. The key goal is to find the optimal regularization parameter, λ using bias-variance decomposition.

**1. Polynomial Regression (Without Regularization)**

In this part of the task, we trained polynomial regression models with degrees ranging from 1 to 9 using **batch gradient descent** for 500 iterations. The goal was to minimize the mean squared error (MSE) between the predicted and actual values.

The graph below illustrates the mean squared error (MSE) as the models iterate over the dataset:



A graph with a line

Description automatically generated

From the plot, we observe the following:

* **Convergence Behaviour**: For all polynomial degrees (1 through 9), the models converge, with MSE gradually decreasing and stabilizing over time.
* **Degree Impact**: As the degree of the polynomial increases, the model becomes more flexible, with lower MSE after convergence. Higher-degree polynomials (e.g., degree 8 and 9) capture the dataset's complexity more effectively, leading to smaller MSE values. However, these models might overfit if unchecked.
* **Key Observation**: While higher-degree polynomials perform better on the training set (lower MSE), they are more prone to overfitting due to the increased flexibility, especially without regularization.

A black background with white and green text

Description automatically generated

This function creates the polynomial feature matrix for the input data X by calculating powers of X up to the given degree (deg). The matrix is then transposed to be used in polynomial regression.

A screen shot of a computer code

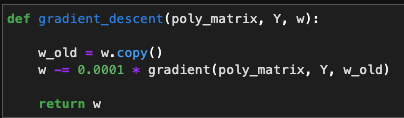
Description automatically generated

This function calculates the mean squared error (MSE) between the predicted values (using the current weights w) and the actual values Y. It sums the squared differences and normalizes by the number of data points by dividing with N.

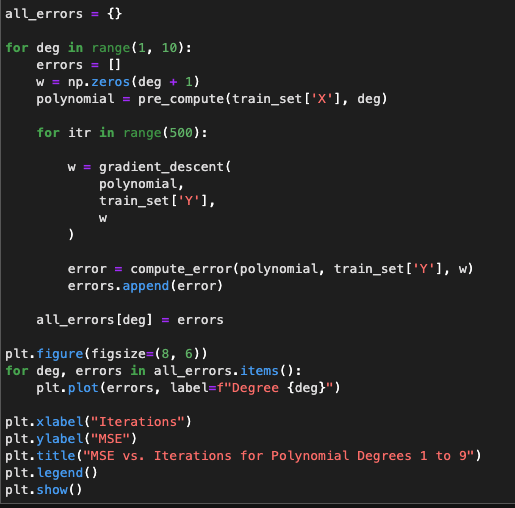
A black background with white text

Description automatically generated

This function computes the gradient of the cost function with respect to the weights w, which represents the slope of the error surface. It uses the difference between predictions and actual values (Y) to update the weights.



Implements the gradient descent algorithm, where the weights w are updated iteratively by moving in the direction of the negative gradient. The learning rate is set to a small value (0.0001) to ensure gradual convergence.



This code trains polynomial regression models for degrees 1 to 9 using **batch gradient descent**. For each degree, it initializes the weights w, generates the polynomial features, and runs 500 iterations of gradient descent to update the weights. The mean squared error (MSE) is computed at each iteration and stored. The plot visualizes the MSE versus iterations for all polynomial degrees, showing the convergence behaviour of each model over time.

**2. Polynomial Regression With Regularization**

In this we have to build 9 new models with L2(Ridge Regression).

We have implemented these functions to work with the extra lambda term introduced by L2 ridge regularization.

A screenshot of a computer program

Description automatically generated

**gradient\_l2 function:**

* This function takes in polynomial (poly\_matrix, Y(actual values), w (weight vector), l2\_lambda (lambda value).
* This function Calculates the gradient taking into account regularization term and adds it to the slope

**gradient\_decent\_l2 function:**

* This function takes in polynomial (poly\_matrix), Y(actual values), w (weight vector), l2\_lambda (lambda value).
* This function is similar to gradient descent function but calculates gradient for a particular lambda.

A screen shot of a computer code

Description automatically generated

compute\_error\_val function: (Works with validation set to get Variance)

* This function takes in polynomial (poly\_matrix), Y(actual values), w (weight vector).
* This function calculates the error of the validation set.

A screenshot of a computer program

Description automatically generated

Explanation of this part of the code:

This code computes bias , variance , total errors and stores it in corresponding dictionaries, where they are mapped to corresponding degrees.

polynomial1 creates the polynomial feature matrix for the training set

polynomial2 creates the polynomial feature matrix for the validation set.

Bias is calculated as the mean squared error of the training set.

Variance is calculated as cross-validation error of the validation set, which is its mean squared error.

For each value of λ (regularization strength), the model is trained using 500 iterations of gradient descent with L2 regularization (gradient\_descent\_l2).

Bias-Variance Trade-off with respect to Lambda

A graph of different numbers

Description automatically generated with medium confidence

The graphs show how different polynomial degrees respond to regularization, with λ controlling the balance between bias and variance. Lower-degree models (1-3) show less improvement with varying λ, while higher degrees (7-9) significantly benefit from optimal λ values to reduce overfitting. Degree 4 achieves the best balance, with regularization further fine-tuning the performance.

We track the overall lowest total error across all degrees and its corresponding lambda. The results are printed below for each degree, along with the degree and lambda that yield the least overall error.

Code:

A computer screen shot of a program

Description automatically generated

Optimal Lambdas:

A computer screen shot of a error

Description automatically generated

Task-3 Visualization of Results

PLOT-1

A graph with blue and orange lines

Description automatically generated

For the regularized polynomial regression curves, we computed and stored the **mean squared error** for both the training and testing sets at each degree. Finally, we plotted the **training and testing errors** against the polynomial degree to visualize the model's performance.

Key Observations

* **Training Error Decreases with Degree**: The training error consistently decreases as the polynomial degree increases, showing that more complex models fit the training data better.
* **Test Error Plateaus**: The test error decreases until around **degree 4–6**, where it stabilizes. Beyond this, increasing the degree further does not improve generalization and might lead to overfitting.
* **Overfitting Trend**: For higher degrees (7–9), the gap between the training and test errors widens, indicating overfitting, where the model fits the noise in the training data instead of the underlying pattern.

PLOT-2

A graph of different colored lines

Description automatically generated

We plotted the bias-variance trade-off for different polynomial degrees by visualizing bias², variance, and total error across various regularization values (log(lambda)). Each degree's error components are shown with distinct line styles for comparison. The plot highlights how regularization impacts the bias-variance balance and total error as lambda changes.

The graph of bias-variance trade-off versus λ shows how different polynomial degrees respond to changes in the regularization parameter:

1. **Bias Trends:** As λ increases, bias gradually rises for all degrees, with low-degree polynomials (1-3) having consistently higher bias. This indicates that stronger regularization reduces model complexity, making it harder to capture the data's structure.
2. **Variance Trends:** Variance decreases as λ increases, with higher-degree polynomials (7-9) showing a significant drop. Regularization helps control variance effectively in more complex models, reducing the risk of overfitting.
3. **Optimal Trade-off:** The middle range of λ values helps achieve a balance between bias and variance, particularly for degree 4, where both are kept relatively low, indicating optimal model performance.

PLOT-3

A graph of a function

Description automatically generated with medium confidence

**We get this curve by selecting the best Lambda(0.278255..) which we get by minimising the total error = bias^2+variance. We plot with degree 4 as it is the appropriate graph by looking at testing error and also because the difference between training and testing error stabilises in plot 1.**   
  
We selected the best polynomial degree (4) based on performance, then precomputed the design matrix for the training set. Using L2-regularized gradient descent, we optimized the weight vector ‘w\_best’ for this degree and the corresponding regularization parameter. Finally, we plotted the fitted curve against the training data to visualize the best polynomial fit.

The degree 4 polynomial was selected because it strikes a balance between capturing the data's non-linear trend and avoiding overfitting. Regularization penalized higher-degree models, ensuring a smoother, generalizable curve with minimized testing error, while lower-degree models underfit the data, failing to capture its complexity.

PLOT-4

A diagram of different colors

Description automatically generated with medium confidence

The code computes MSE for polynomial models of different degrees, varying learning rates, and regularization strengths using L2-regularized gradient descent. It then visualizes the MSE surfaces for each degree with 3D plots.

* X-axis represents the Learning Rate.
* Y-axis represents the Lambda (regularization strength).
* Z-axis shows the Mean Squared Error (MSE).

Each plot corresponds to a different polynomial degree (ranging from 1 to 9 in your case), showing how model complexity interacts with regularization and learning rate to affect MSE.

* **Lower Degrees (e.g., 1, 2, 3):** These tend to underfit the data, so the MSE is higher overall.
* **Medium Degree (e.g., 4, 5):** For many datasets, these provide a good balance between bias and variance, leading to lower MSE in certain regions.
* **Higher Degrees (e.g., 7, 8, 9):** These may overfit the data, resulting in larger errors unless strong regularization is applied (indicated by higher lambda values).
* The plots show how **MSE changes** as you adjust both the **learning rate** and **regularization strength** for different degrees of the polynomial.
* For higher degrees, more regularization (higher lambda) is needed to prevent overfitting.
* For most degrees, there is a region where the combination of a moderate learning rate and appropriate lambda yields the lowest MSE, indicating the best model performance.

Task-4 Comparative Analysis

**Q1)Perform a comparative analysis of the polynomial regression models with and without regularization.**

We need to examine the effects of adding regularization at different polynomial degrees. Here’s an analysis based on the provided results:

**Without Regularization (Basic Polynomial Regression):**

**1. Degree 1 to 3:**

- The mean squared error (MSE) starts relatively high and decreases with iterations. However, there is a limit to the improvement as the polynomial degree increases.

- The bias remains high across these degrees due to underfitting since the models cannot capture the complexity of the data.

- Variance is low due to the simplicity of the model, resulting in lower total error.

**2. Degree 4 to 6:**

- The MSE improves significantly as the model becomes more capable of fitting the data.

- The bias is reduced, but the variance starts to increase because of the higher degree polynomial capturing more noise.

- Training error is relatively low for degree 4, indicating a good fit without overfitting. However, higher degrees (5 and 6) show signs of good enough fitting.

**3. Degree 7 to 9:**

- The variance dominates over bias due to the high model complexity.

- The MSE shows further reductions but at the cost of very high variance, making these models sensitive to noise.

- The total error is higher due to the high variance, reflecting overfitting behavior.

**With Regularization (Ridge Regression):**

**1. Degree 1 and 2 (Optimal Lambda = 2.15):**

- The optimal lambda for lower degrees is higher, indicating more regularization needed due to simplicity.

- Regularization effectively reduces the variance, leading to a slightly lower total error compared to the models without regularization.

- The bias remains higher due to underfitting.

**2. Degree 3 to 6 (Optimal Lambda = 0.28):**

- The regularization reduces the variance, balancing the bias-variance trade-off better than without regularization.

- Lower lambda values are needed as the degree increases since the model is already more complex.

- The total error is minimized at these degrees due to an effective reduction in variance without significantly increasing bias.

**3. Degree 7 to 9 (Optimal Lambda = 0.036):**

- The smallest lambda values are used here because the model complexity is already very high.

- Regularization helps control the variance, significantly reducing overfitting compared to unregularized models.

- For degree 9, the overall least total error (0.2999) is achieved, indicating a well-regularized model despite the high polynomial degree.

Summary:

**Without regularization**

* The higher-degree polynomial models exhibit overfitting due to high variance. Lower degrees suffer from underfitting as they cannot capture the data complexity.

**With regularization**

* The models achieve a better balance between bias and variance. The optimal lambdas vary based on the degree, with more regularization required for simpler models and less regularization for more complex models.

Degree 4 provides the best fit without regularization due to a good balance between training and testing errors. With regularization, degree 9 achieves the lowest total error, showcasing the benefit of ridge regression in controlling overfitting for high-degree polynomials.

The visualizations confirm these trends, showing reduced variance with increasing regularization strength (lambda), leading to lower total error across polynomial degrees.

**Q2) Discuss the behaviour of the model as you increase the polynomial degree and how**

**regularization helps mitigate overfitting.**

Regularization  
1) **Reduces Variance**:

* Regularization, such as ridge regression, adds a penalty term to the loss function that discourages large coefficients. This effectively controls the model's complexity and reduces the magnitude of the fitted parameters.
* In high-degree polynomials (7-9), where variance dominates, regularization helps lower the variance without significantly increasing bias, making the model's predictions more stable.

2) **Balances Bias and Variance Trade-off**:

* By tuning the regularization parameter (lambda), the model can achieve an optimal balance between bias and variance. For instance, smaller lambda values for higher-degree polynomials allow enough flexibility to fit the data while keeping the model from overfitting.
* For degree 4, regularization fine-tunes the fit by slightly reducing variance, which helps maintain a low testing error, further supporting its status as the best-fitting model.

3) **Improves Generalization**:

* Regularization prevents the model from becoming too specific to the training data by smoothing the decision boundary. This leads to better performance on unseen data, thus improving generalization.
* The plots of bias, variance, and total error across different lambdas show that regularization helps keep the total error low, especially for complex models where the risk of overfitting is higher.

As polynomial degree increases, regularization becomes crucial for controlling the model's complexity. It mitigates overfitting by reducing variance and ensures that the model performs well on testing data, especially for high-degree polynomials where overfitting is more pronounced. Degree 4 remains a good choice with or without regularization due to its balanced performance on both training and testing sets.

**Q3) Analyse how the bias and variance change as you move from underfitting to overfitting polynomials.**

As the polynomial degree increases, the bias-variance behaviour changes from underfitting to overfitting:

1. **Underfitting (Low-Degree Polynomials, e.g., 1-3):**

- High Bias: The model is too simple to capture the data's underlying patterns, leading to systematic prediction errors. It assumes the relationship between features and the target is less complex than it actually is.

- Low Variance: Predictions are stable across different datasets, as the model does not react strongly to changes in the training data.

2**. Good Fit (Moderate-Degree Polynomials, e.g., 4-6):**

- Reduced Bias: The model becomes more flexible, capturing the data's structure better, and prediction errors due to oversimplification decrease.

- Moderate Variance: The model starts to pick up more details, but variance remains controlled, resulting in good generalization to new data.

3. **Overfitting (High-Degree Polynomials, e.g., 7-9).**

- Low Bias: The model fits the training data very well, even capturing noise, resulting in minimal prediction errors on the training set.

- High Variance: The model's predictions vary significantly with small changes in the training data, leading to poor generalization and high testing error due to sensitivity to noise.

Thus, moving from underfitting to overfitting involves a trade-off between decreasing bias and increasing variance.

**Q4) Report your observations on how the regularization parameter λ affects the model’s complexity and performance.**

The bias-variance plots with regularization show how the regularization parameter, λ, affects the model's complexity and performance across different polynomial degrees:

**1. Low λ Values (Near Zero):**

- The model is minimally regularized, allowing high complexity. This leads to low bias but high variance as the model can fit the training data very closely, including noise. For higher polynomial degrees (7-9), the total error increases due to overfitting, which is evident from the sharp rise in variance.

**2. Moderate λ Values:**

- As λ increases, regularization becomes stronger, which helps reduce the variance by penalizing large coefficients. This smoothing effect lowers the model's flexibility, resulting in a balanced bias-variance trade-off.

- For moderate degrees (4-6), the total error is minimized at an optimal λ, indicating a balance where the bias and variance are both reasonably controlled, leading to the best generalization performance.

**3. High λ Values:**

- With large λ values, regularization heavily penalizes model complexity, leading to increased bias and lower variance. The model becomes overly simplistic, failing to capture essential data patterns, thus causing underfitting. This is especially clear for low-degree polynomials (1-3), where the total error spikes due to high bias.

**Observations:**

- Effect on Complexity: Increasing λ simplifies the model by constraining its ability to fit the data, reducing complexity as λ grows.

- Effect on Performance: Proper tuning of λ helps mitigate overfitting by lowering variance and controlling the total error. However, excessively large λ values can lead to underfitting by making the model too rigid.

Overall, the regularization parameter λ helps adjust the bias-variance trade-off, allowing the model to generalize better by avoiding both overfitting and underfitting.

**Observations on model performance, overfitting, underfitting, and regularization.**

**Model Performance:**

* improves with increasing polynomial degree until around degree 4, where the testing error is minimized, indicating a good fit. Higher degrees lead to overfitting, with low training error but increased testing error due to high variance. Regularization helps control this by reducing variance and enhancing generalization, especially in complex models.

**Underfitting (Low Degrees, 1-3):**

* The models show high bias and low variance, indicating underfitting, as they cannot capture the data's complexity. Even with regularization, the total error remains relatively high due to insufficient model flexibility.

**Best Fit (Degree 4):**

* Degree 4 achieves the best balance between bias and variance, resulting in the lowest testing error without regularization. Regularization fine-tunes the model by slightly reducing variance, leading to stable performance.

**Overfitting (High Degrees, 7-9):**

* Without regularization, the models exhibit low bias but very high variance, leading to overfitting as they fit noise in the training data. The testing error is much higher than the training error.
* Regularization helps control the variance in these higher-degree models, reducing total error and improving generalization.

**Effect of λ (Regularization Parameter):**

* Increasing λ reduces model complexity by controlling the magnitude of coefficients, which lowers variance and mitigates overfitting. However, very high λ values increase bias, leading to underfitting, as seen in high total error at large λ values.