Linear and Non-Linear Regression

Angello Soldi* ANGELLO.SOLDI@UTEC.EDU.PE
UTEC, Peru

Fabian Castro* FABIAN.CASTRO@UTEC.EDU.PE
UTEC, Peru

Abstract

This study analyzes a time series of climate measurements by applying linear and nonlinear regression, with and without L1 (Lasso) and L2 (Ridge) regularization. The performance of the models is evaluated using the mean squared error (MSE) to identify the most accurate approach. The results compare the effectiveness of each technique in predicting climate data.

Keywords: Regression, Regularization, MSE

1. Introduction

This paper aims to analyze a one-dimensional time series of climate measurements over time through the implementation of regression techniques, evaluating both linear and nonlinear models. The objective is to identify which of these approaches is better suited to capture the inherent trends in the data and provide accurate predictions for future measurements. To achieve this, experiments will be conducted comparing models without regularization and those with L1 (Lasso) and L2 (Ridge) regularization, in order to analyze the impact of each method on reducing model complexity, mitigating overfitting, and improving generalization capacity. The mean squared error (MSE) will be used as the evaluation metric to measure the accuracy of the predictions relative to the actual values. The results obtained will be analyzed to establish comparisons between the different approaches, thereby determining which offers the best performance in the specific context of the climate time series under study.

Methodology

In this study, two types of regression models will be implemented to analyze the time series of climate data: one linear and one sinusoidal. Both models will be evaluated in terms of their ability to fit the data and make accurate predictions. To determine the best model, we will use the Mean Squared Error (MSE) as the evaluation metric.

- **Linear Regression Model:** This model will be applied both without regularization techniques and with L1 (Lasso) and L2 (Ridge) regularization techniques. Regularization helps control the complexity of the model, preventing overfitting, especially when there is variability in the data.

- **Sinusoidal Model:** Given that the data exhibits a cyclical behavior, a sinusoidal regression model will be used to capture this pattern. This model will be tested both without regularization and with the same L1 and L2 regularization techniques.

The comparison between the two models will be based on the MSE values, with the goal of identifying which of the two approaches provides a better fit to the data and is more effective at predicting future measurements. The model with the lowest MSE value will be considered the most suitable for this dataset.

2. Data Exploration

First, a visual analysis of the time series data of climate measurements was conducted by creating a line chart. The horizontal axis (X) represents the days, while the vertical axis (Y) shows the recorded daily temperatures. The chart reveals a cyclical pattern, suggesting periodic variations in temperatures over time. This type of behavior may be related to seasonal changes or daily fluctuations in weather conditions. Temperatures range between approximately 20°C and 80°C, indicating a significant amplitude in the measurements.

* These authors contributed equally

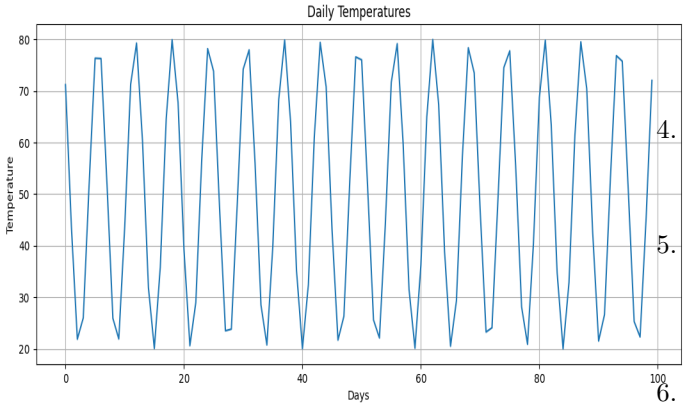

Figure 1: Daily Temperature Chart.

The initial analysis of the chart reveals a tendency to repeat regular cycles, suggesting that a nonlinear regression model might be more suitable for capturing this type of periodic behavior. However, it is also possible to explore a linear regression model as a reference to evaluate its fitting capability against this time series.

2.1. Basic Statistics

Statistic	Value
Count	100
Mean	49.835852
Standard Deviation	21.255238
Min	20.021208
25% Percentile	28.415242
50% Percentile	49.867323
75% Percentile	71.236622
Max	79.967635

Table 1: Descriptive statistics of the temperature data.

1. **Count (100):** This represents the total number of temperature observations in the dataset, which in this case is 100.
2. **Mean (49.84):** The mean represents the average temperature value across all observations, which is approximately 49.84°C.
3. **Standard Deviation (21.25):** The standard deviation quantifies the dispersion of the data relative to the mean. A value of 21.25°C indicates considerable variability in the temperatures.
4. **Minimum (20.02):** The minimum recorded temperature is 20.02°C, representing the lowest value observed in the dataset.
5. **25th Percentile (28.41):** This value indicates that 25% of the temperatures are below 28.41°C, providing information about the lower range of the data distribution.
6. **50th Percentile (Median, 49.87):** The median, which is 49.87°C, represents the midpoint of the dataset, where half of the temperatures are lower and the other half are higher. It is close to the mean, suggesting a relatively symmetric distribution.
7. **75th Percentile (71.24):** The 75% percentile indicates that 75% of the data falls below 71.24°C, offering insight into the upper range of the temperature distribution.
8. **Maximum (79.97):** The maximum recorded temperature is 79.97°C, being the highest value in the dataset.

2.2. Rolling Mean and Standard Deviation

The following graph displays the moving average and moving standard deviation statistics of the temperature measurements. Below are the interpretations of each of the lines:

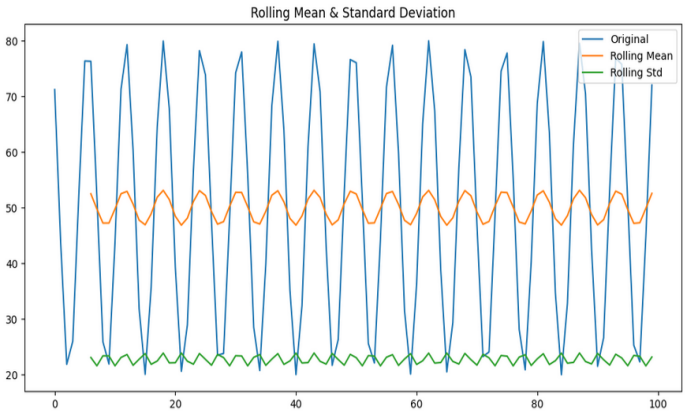

Figure 2: Graph of the Moving Average and Moving Standard Deviation of Temperature.

1. **Original Series (blue line):** The blue line represents the original time series of temperatures. A clearly defined cyclical pattern is observed, with significant fluctuations between 20°C and 80°C at regular intervals. This indicates that the data have a periodic component, likely influenced by seasonal or daily cycles.

2. **Moving Average (orange line):** The moving average smooths short-term fluctuations in the data and shows the overall trend in the temperature behavior over time. The moving average follows a regular trajectory, oscillating around 50°C, which suggests that the significant fluctuations in the time series remain within a stable range, without a clear long-term upward or downward trend.

3. **Moving Standard Deviation (green line):** The moving standard deviation reflects the volatility of the temperature measurements within a given time interval (in this case, 7 days). The green line remains quite low and stable, indicating that the short-term variability within each 7-day window is relatively small compared to the variability observed in the original series. This reinforces the idea that the series is cyclical and does not show significant changes in variation over time.

Conclusions: The cyclical behavior of the data is confirmed both in the original series and in the moving average. The absence of significant changes in the moving standard deviation suggests that the periodic cycle is quite regular, with little variability between cycles. This analysis is useful for predicting future measurements and adjusting regression models, especially if we aim to capture these cyclical patterns.

2.3. Important Graphs

2.3.1. Histogram

The following graph displays a histogram of the temperatures with an overlaid kernel density estimation (KDE) curve.

A bimodal distribution is observed, with peaks at the extremes near 20°C and 80°C, indicating a higher concentration of temperatures in those ranges. The frequencies of intermediate temperatures (30°C - 70°C) are lower. The KDE curve follows the shape of the histogram and confirms the bimodal nature of the data. This suggests that

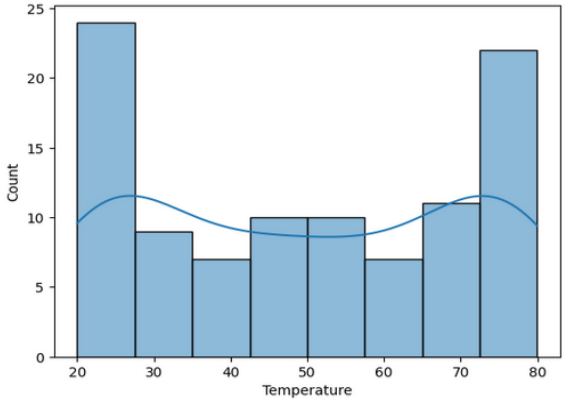

Figure 3: Histogram

temperatures tend to cluster around two key points, which may be related to different climatic conditions.

Conclusion: The bimodal distribution implies that a simple linear model may not be sufficient to capture the complexity of these data. It is recommended to consider this characteristic when selecting a predictive model.

2.3.2. Q-Q Plot

The Q-Q (Quantile-Quantile) plot compares the quantiles of the observed data distribution with those of a theoretical normal distribution.

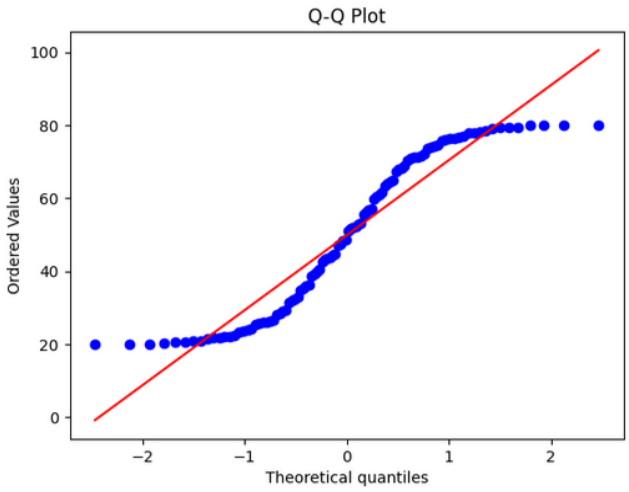

Figure 4: Q-Q Plot

1. **Deviation from the theoretical line:** The observed points deviate from the reference red line, especially in the tails, indicating that the data do not follow a normal distribution.

2. **S-shape:** The S-shaped curve suggests a bimodal distribution, confirming the previous analysis from the histogram and KDE.

Conclusion: The Q-Q plot demonstrates that the temperature data do not follow a normal distribution, which implies that models assuming normality may not be suitable for these data.

3. Experimentation

3.1. Linear Regression

Derivation Rules Used in the Algorithm Implementation

In the gradient descent linear regression algorithm, basic rules of differentiation are used to update the parameters m (slope) and c (intercept) of the regression line. Below are the derivatives of the cost function with respect to m and c.

The cost function being minimized is the **Mean Squared Error (MSE)**, defined as:

$$J(m, c) = \frac{1}{N} \sum_{i=1}^{N} (y_i - (mx_i + c))^2$$

Where:

- N is the number of observations.
- y_i is the actual value of the dependent variable.
- x_i is the value of the independent variable.
- m is the slope (coefficient of the independent variable).
- c is the intercept or constant term.

Derivative of $J(m, c)$ with respect to m

To find the gradient of the cost function with respect to m, we differentiate $J(m, c)$ with respect to m:

$$\frac{\partial J}{\partial m} = \frac{1}{N} \sum_{i=1}^{N} -2x_i(y_i - (mx_i + c))$$

That is, the gradient with respect to m is calculated as:

$$\text{gradient\_m} = -\frac{2}{N} \sum_{i=1}^{N} x_i(y_i - (mx_i + c))$$

Derivative of $J(m, c)$ with respect to c

Similarly, we differentiate $J(m, c)$ with respect to c:

$$\frac{\partial J}{\partial c} = \frac{1}{N} \sum_{i=1}^{N} -2(y_i - (mx_i + c))$$

Therefore, the gradient with respect to c is calculated as:

$$\text{gradient\_c} = -\frac{2}{N} \sum_{i=1}^{N} (y_i - (mx_i + c))$$

How Linear Regression with Gradient Descent Works

Linear regression is a method that seeks to fit a straight line to a set of points, such that the sum of the squared errors between the observed values Y and the values predicted by the model Y_{pred} is minimized. This line is defined by the equation:

$$y = mx + c$$

Where m is the slope of the line and c is the intercept on the y-axis. The goal is to find the optimal values of m and c that minimize the cost function (MSE).

Gradient Descent Algorithm Process

1. **Initialization**: We start with initial values for m and c, usually 0.

2. **Error Calculation**: The error is calculated between the predicted values $y_{\text{pred}} = mx + c$ and the actual values y.

3. **Gradient Calculation**: The derivatives explained earlier are used to calculate the gradients of the cost function with respect to m and c.

4. **Parameter Update**: m and c are updated at each iteration by subtracting the gradients multiplied by the learning rate α:

$$m = m - \alpha \frac{\partial J}{\partial m}$$

$$c = c - \alpha \frac{\partial J}{\partial c}$$

5. **Repeat**: This process is repeated for a number of iterations, until the gradients approach zero or until a maximum number of iterations is reached. As the gradients reduce, the values of m and c converge to the optimal values that minimize the error.

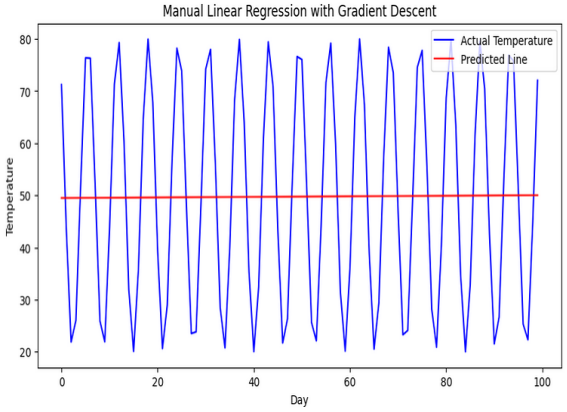

Figure 5: Linear Regression

Conclusion

Based on the analysis of the obtained graph and the Mean Squared Error (MSE) value of 437.2914, the following conclusions can be drawn:

a) **Misfit of the linear model**: The obtained regression line is a horizontal straight line that does not adequately capture the variability of the data, which follows an oscillating pattern. This indicates that the linear model is not suitable for representing the relationship between the days and the temperature.

b) **High error**: The MSE value of 437.2914 is considerably high, reflecting that the difference between the observed values (actual temperatures) and the values predicted by the model is large. This reinforces the idea that the linear model is not effective for these data.

c) **Periodic nature of the data**: The behavior observed in the data shows a clear periodic oscillation that cannot be captured by linear regression. This type of cyclic behavior suggests that a sinusoidal function or a non-linear model would be more appropriate.

d) **Need for a non-linear model**: Since the periodic nature of the data cannot be modeled with linear regression, a better alternative would be to use a sinusoidal model that captures the inherent periodicity of the data and provides more accurate predictions.

e) **Final conclusion**: Linear regression, being a simple model limited to linear relationships, is not appropriate for this dataset. A more complex approach, such as sinusoidal fitting, is required to reduce the MSE and provide a better fit to the observed data.

3.2. Sinusoidal Regression

Sinusoidal Model Algorithm

This algorithm fits a sinusoidal model to a dataset using the gradient descent method. The sinusoidal model follows the form:

$$Y = a \cdot \sin(bX + c) + d$$

Where a, b, c, and d are the parameters that the algorithm adjusts to minimize the mean squared error (MSE) between the observed and predicted values.

Sinusoidal Model Function

The sinusoidal model is defined as:

$$\text{sinusoidal}(X, a, b, c, d) = a \cdot \sin(bX + c) + d$$

Where X is the input vector, and a, b, c, and d are the parameters to be adjusted.

Cost Function (MSE)

The loss function used is the Mean Squared Error (MSE), defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_{\text{actual}} - Y_{\text{predicted}})^2$$

Where:

- Y_{actual} are the observed data values.
- $Y_{\text{predicted}}$ are the values predicted by the sinusoidal model.

Gradient Descent for Sinusoidal Parameters

Gradient descent is used to minimize the MSE by updating the parameters a, b, c, and d. To do this, the partial derivatives of the MSE with respect to each parameter are calculated.

Partial Derivatives

The partial derivatives of the cost function with respect to each parameter are:

$$\frac{\partial MSE}{\partial a} = -\frac{2}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i) \cdot \sin(bX_i + c)$$

$$\frac{\partial MSE}{\partial b} = -\frac{2}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i) \cdot aX_i \cdot \cos(bX_i + c)$$

$$\frac{\partial MSE}{\partial c} = -\frac{2}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i) \cdot a \cdot \cos(bX_i + c)$$

$$\frac{\partial MSE}{\partial d} = -\frac{2}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)$$

Where:

- n is the number of data points.
- X_i is the value of the independent variable.
- Y_i is the observed value.
- $\hat{Y}_i$ is the predicted value by the model.

Parameter Update

The parameters are updated in each iteration using the following formulas:

$$a = a - \alpha \cdot \frac{\partial MSE}{\partial a}$$

$$b = b - \alpha \cdot \frac{\partial MSE}{\partial b}$$

$$c = c - \alpha \cdot \frac{\partial MSE}{\partial c}$$

$$d = d - \alpha \cdot \frac{\partial MSE}{\partial d}$$

Where α is the learning rate.

Gradient Descent Implementation

The gradient descent algorithm adjusts the parameters a, b, c, and d by minimizing the MSE. In each iteration, the parameters are updated based on the gradients.

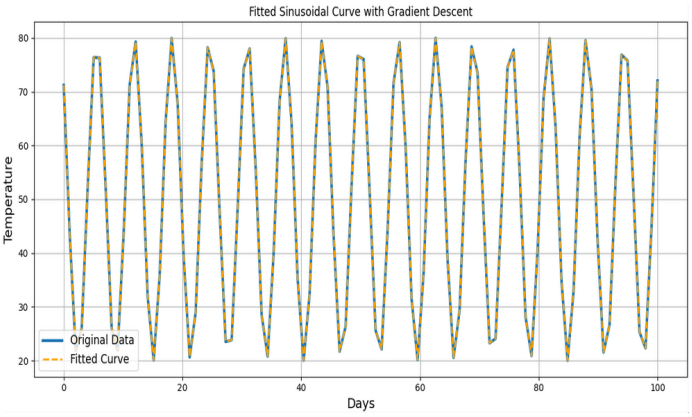

Figure 6: Sinusoidal Regression

Conclusion

Based on the sinusoidal fitting performed on the temperature data and the obtained graph, the following conclusions can be drawn:

a) **Good fit of the sinusoidal model**: The adjusted sinusoidal model (orange dashed line) closely follows the actual data values (blue line). This indicates that the sinusoidal model has successfully captured the periodic pattern of the data.

b) **Capture of periodicity**: The sinusoidal fit accurately reflects the cyclic nature of the data. The maxima and minima of the fitted curve align well with those of the original data, suggesting that the model has correctly captured the frequency of the cycles.

c) **Convergence with gradient descent**: The gradient descent algorithm has successfully found the optimal parameters a, b, c, and d that best describe the data. Despite using a large number of iterations (epochs) and a low learning rate, the algorithm has converged to a precise solution, minimizing the mean squared error (MSE).

Sinusoidal Regression Regularization

L1 Regularization

L1 regularization, also known as **Lasso** (*Least Absolute Shrinkage and Selection Operator*), is a technique that adds a penalty term based on the *sum of the absolute values* of the model coefficients. The goal of this regularization is to prevent overfitting and to shrink some coefficients to zero, leading to a simpler model.

L1 Regularization Formula:

$$\text{Total Cost} = \text{MSE} + \alpha_{\text{L1}} \sum_i |w_i|$$

Where:

- MSE is the *Mean Squared Error*, which measures the difference between predicted and actual values.
- α_{L1} is the regularization hyperparameter, which controls the magnitude of the penalty.
- w_i are the model coefficients.

The term $\alpha_{\text{L1}} \sum_i |w_i|$ is known as the **L1 regularization term**, and its main effect is that it reduces some of the model's coefficients to zero when the penalty is high, resulting in a more sparse or parsimonious model.

Application in Code: In the code, L1 regularization is incorporated by modifying the loss function with a penalty term. The regularization term α_{L1} is added to the *MSE* loss, and it affects the gradient values of the parameters.

The coefficients a, b, c, and d are adjusted in each iteration using gradient descent, with an additional term that reduces their magnitude based on α_{L1} and the sign of the parameter.

$$\text{MSE}_{L1} = \text{MSE} + \alpha_{\text{L1}} \left(|a| + |b| + |c| + |d|\right)$$

The partial derivatives of the cost function with L1 regularization with respect to each parameter are:

$$\frac{\partial \text{MSE}_{L1}}{\partial a} = \frac{\partial \text{MSE}}{\partial a} + \alpha_{\text{L1}} \cdot \text{sign}(a)$$

$$\frac{\partial \text{MSE}_{L1}}{\partial b} = \frac{\partial \text{MSE}}{\partial b} + \alpha_{\text{L1}} \cdot \text{sign}(b)$$

$$\frac{\partial \text{MSE}_{L1}}{\partial c} = \frac{\partial \text{MSE}}{\partial c} + \alpha_{\text{L1}} \cdot \text{sign}(c)$$

$$\frac{\partial \text{MSE}_{L1}}{\partial d} = \frac{\partial \text{MSE}}{\partial d} + \alpha_{\text{L1}} \cdot \text{sign}(d)$$

Where:

- α_{L1} is the L1 regularization hyperparameter.
- $\text{sign}(x)$ is the sign function, which returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.

L2 Regularization

L2 regularization, also known as **Ridge**, adds a penalty term based on the *sum of the squares* of the model coefficients. Unlike L1, L2 does not force the coefficients to be zero but instead reduces their values, stabilizing the model and preventing overfitting.

L2 Regularization Formula:

$$\text{Total Cost} = \text{MSE} + \alpha_{\text{L2}} \sum_i w_i^2$$

Where:

- MSE is the *Mean Squared Error*.
- α_{L2} is the L2 regularization hyperparameter, which controls the weight of the penalty.
- w_i are the model coefficients.

The L2 regularization term $\alpha_{\text{L2}} \sum_i w_i^2$ penalizes large values of the coefficients and tends to reduce their magnitude, making the model smoother. However, unlike L1, it does not bring the coefficients to zero, so all features remain active.

Application in Code: Similar to L1, L2 regularization is added to the loss function and affects the gradients of the coefficients. The coefficients a, b, c, and d are adjusted with an additional term $2\alpha_{\text{L2}} w_i$, which reduces the magnitude of the coefficients in proportion to their current value. Both regularizations are applied during the training of the sinusoidal model using gradient descent.

$$\text{MSE}_{L2} = \text{MSE} + \alpha_{\text{L2}} \left(a^2 + b^2 + c^2 + d^2\right)$$

The partial derivatives of the cost function with L2 regularization with respect to each parameter are:

$$\frac{\partial \text{MSE}_{L2}}{\partial a} = \frac{\partial \text{MSE}}{\partial a} + 2\alpha_{\text{L2}} a$$

$$\frac{\partial \text{MSE}_{L2}}{\partial b} = \frac{\partial \text{MSE}}{\partial b} + 2\alpha_{\text{L2}} b$$

$$\frac{\partial \text{MSE}_{L2}}{\partial c} = \frac{\partial \text{MSE}}{\partial c} + 2\alpha_{\text{L2}} c$$

$$\frac{\partial \text{MSE}_{L2}}{\partial d} = \frac{\partial \text{MSE}}{\partial d} + 2\alpha_{\text{L2}} d$$

Where:

- α_{L2} is the L2 regularization hyperparameter.

Comparison Plot

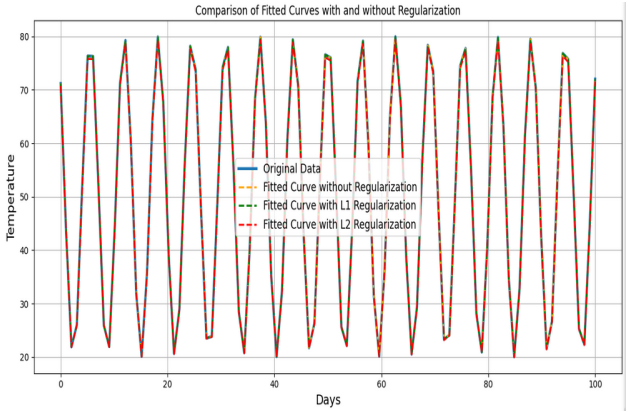

Figure 7: Sinusoidal Regression Regularization

4. Discussion

5. Discussion

Relationship between the L1/L2 Code and Machine Learning

The code presented applies L1 and L2 regularization in a sinusoidal regression model using gradient descent. Below, some relevant theoretical concepts are related to what happens in the code.

Without Regularization

The code adjusts a sinusoidal model to the temperature data without applying regularization, meaning that the model has complete freedom to fit the training data. This fit can lead to **overfitting**, as the model has the capacity to capture all the variance in the training data, but may not generalize well to new data. This is reflected in a low mean squared error (**MSE**) during training, but potentially high error on new data.

With L1 and L2 Regularization

The code then introduces L1 (Lasso) and L2 (Ridge) regularization to the sinusoidal model. Regularization penalizes the model parameters, forcing the model to be simpler and reducing its ability to fit the data exactly. This reduces the risk of **overfitting**, but increases the MSE on the training data, indicating lower precision on the adjusted data.

Bias-Variance Tradeoff

Regularization directly affects the **bias-variance tradeoff**. Without regularization, the model has low variance and high bias, which can lead to overfitting. By applying L1 or L2 regularization, the bias increases (the model becomes more rigid), but the variance is reduced, improving the model's ability to generalize to new data. The challenge is to find the optimal point of regularization where the MSE on unseen data is minimized.

Overfitting and Underfitting

This balance is key to avoiding both **overfitting** (where the model is too flexible and fits the noise in the training data) and **underfitting** (where the model is too simple and fails to capture the underlying relationship in the data). Regularization helps avoid overfitting, but if the penalty is too strong, it can lead to underfitting.

6. Citations and Bibliography

References

[1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[2] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

7. Source Code

Repo