ML Documentation

1. Project Description

General Information on Dataset:

- **Dataset Name**: STL-10
- **Number of Classes**: 5
- **Labels**: Airplane, Bird, Car, Cat, Deer.
- **Total Number of Samples**: 100,000 (including unlabeled images)
- **Image Sizes**: 96x96 pixels.
- **Missing Data**: No missing data
- **Dataset Split**:
    - Training Samples: 500 (10 pre-defined folds of 100 examples)
    - Validation Samples: [Not explicitly defined]
    - Testing Samples: 8,000 (800 per class)
    - Unlabeled Samples: 100,000 for unsupervised learning

Description:

The STL-10 dataset is inspired by CIFAR-10 but offers higher resolution images and a larger corpus for unsupervised learning. It consists of 10 classes, with labeled training and testing samples as well as a large pool of unlabeled images for unsupervised feature learning. This dataset is particularly suitable for developing deep learning, self-taught learning, and scalable unsupervised methods.

2. Algorithms Implemented

a. Linear Regression

- **Description**: Linear Regression is a supervised learning algorithm used for predicting continuous values. It models the relationship between independent variables (features) and a dependent variable using a linear equation.

b. Logistsic Regression

Description: Logistic Regression is a supervised learning algorithm used for binary classification tasks. It models the relationship between independent variables (features) and a dependent variable by applying the logistic function to predict probabilities, which are then mapped to two possible outcomes. The model estimates the probability of a given input belonging to a certain class by fitting a linear equation and applying a sigmoid function to produce output between 0 and 1.

c. K-Nearest Neighbors (KNN)

- **Description**: KNN is a non-parametric, instance-based learning algorithm that classifies data points based on the majority class of their nearest neighbors in feature space.

3. Comparison between Numeric And Image Recognition

Feature	K-Nearest Neighbors (KNN)	Linear Algorithms (Linear Regression, Logistic Regression)
Dataset Type	Unsupervised and supervised (classification)	Supervised (classification or regression)
Problem Type	Classification (image recognition of animals/vehicles)	Classification (Logistic Regression) or Regression (Linear Regression)
Training Time	Very slow (no explicit training, stores all data)	Faster (requires training to compute coefficients)
Prediction Time	Slow (distance computation to all training points)	Fast (direct computation using learned coefficients)
Feature Handling	Does not require feature extraction; works directly with raw pixel values	Typically requires feature extraction or preprocessing (e.g., flattening, normalization)
Model Assumptions	No assumptions about data distribution (non-parametric)	Assumes a linear relationship (for Linear Regression and Logistic Regression)
Memory Usage	High (stores all training data)	Low (stores only model coefficients)
Sensitivity to High Dimensionality	Very high (distance calculations are affected by the curse of dimensionality)	Moderate to low (Linear models may require dimensionality reduction techniques)
Sensitivity to Noise/Outliers	High (outliers affect classification decisions)	Moderate (outliers can still impact model accuracy)
Interpretability	Low (hard to understand how the decision is made for new data)	High (model coefficients provide insight into feature importance)
Scalability	Poor (due to memory and computation overhead)	Good (easy to scale with large datasets)

Model Complexity	Simple to implement but computationally expensive	Simple and computationally efficient after training
Handling Multi-Class	Directly handles multi-class classification by selecting the majority label among neighbors	Logistic Regression: Requires modifications like one-vs-rest for multi-class
Accuracy	Can perform well on complex datasets with high-dimensional features like images, especially when features are well-chosen or extracted	Can be limited in accuracy on raw image data, requires preprocessing (e.g., feature extraction)
Example Applications	Image recognition (animals, vehicles)	Logistic Regression: Image classification with feature extraction; Linear Regression: Regression tasks like predicting continuous variables from image features

---

Observations:

- **K-Nearest Neighbors (KNN)**: While KNN can be effective in simpler tasks, it struggles with the **high-dimensionality** of image data in the **STL-10 dataset** (96x96 pixel images). The **curse of dimensionality** impacts the algorithm's performance, especially since it computes distances between each test sample and all training points during prediction. This results in high **memory usage** and **slow prediction times**, making KNN less efficient for large image datasets like STL-10, unless additional **dimensionality reduction** techniques (e.g., PCA) are used beforehand.

- **Linear Algorithms (Logistic Regression)**: Logistic Regression requires feature extraction to perform well on image data. Raw images cannot be directly fed into the model, so techniques like **flattening** or using **pre-trained models** to extract features are necessary. Once features are extracted, Logistic Regression performs well, with **faster training and prediction times** compared to KNN, as it directly computes results using learned coefficients. However, its performance is still limited because it assumes a **linear relationship**, which is inadequate for complex, high-dimensional image data. To achieve better results, the model would require **advanced feature extraction** or deeper learning techniques like Convolutional Neural Networks (CNNs).

Conclusion:

- **KNN** struggles with large, high-dimensional datasets like STL-10 due to computational complexity and memory usage, particularly during prediction. It's not scalable without dimensionality reduction or feature engineering.

- **Logistic Regression**, while more efficient once features are extracted, can be limited in accuracy when working with raw image data. It requires careful preprocessing and may not capture the complex patterns in image data effectively.

Performance Comparison: **Logistic Regression** is generally **faster** and **more scalable** than KNN for the STL-10 dataset, but it requires good feature engineering. **KNN** could perform better in simpler cases where features are already well-defined or reduced but is less effective on raw, high-dimensional image data.

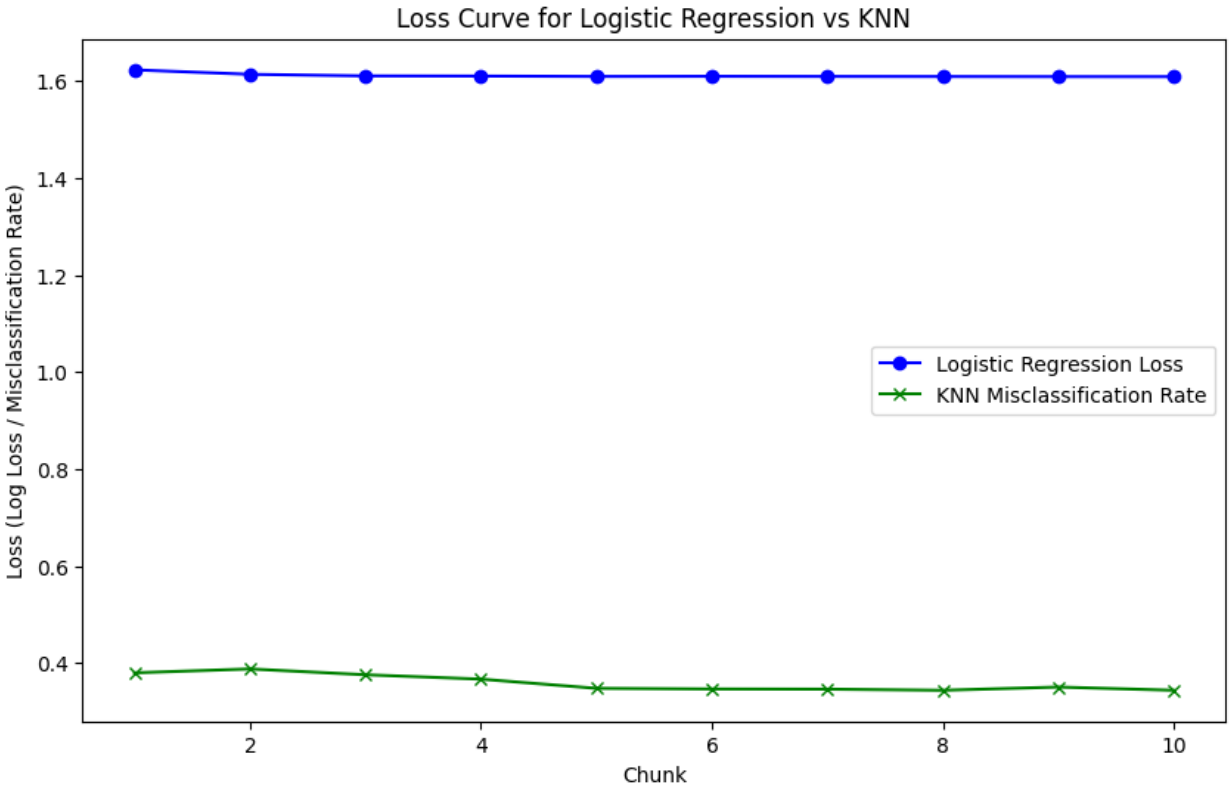
---

4. Implementation Details for Models

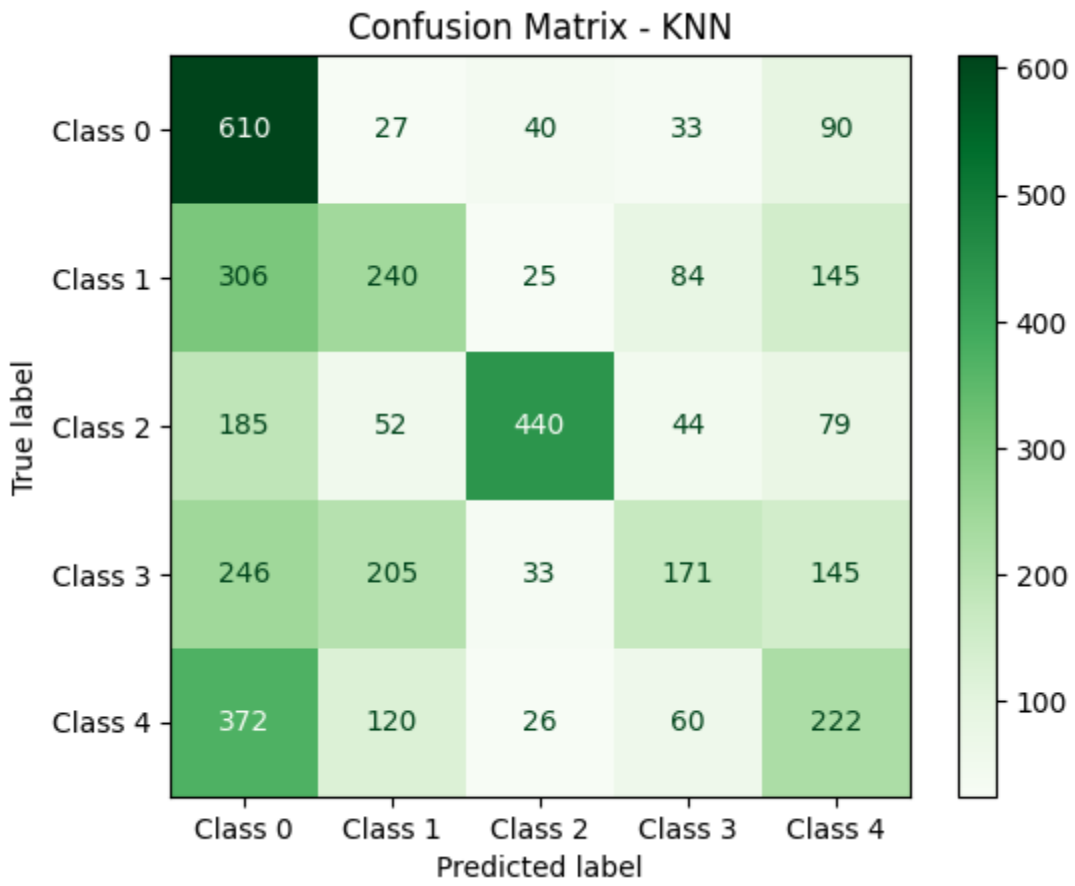
For each model, we evaluated the following on testing data:

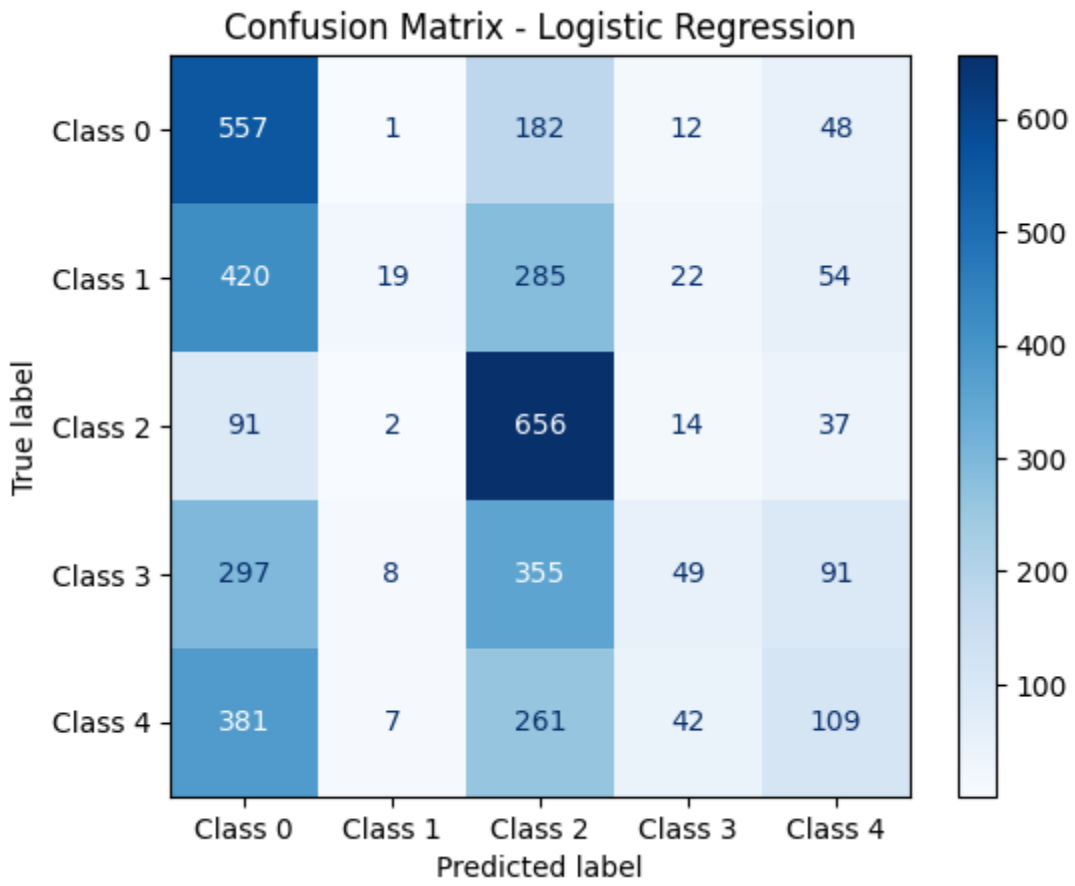
a. Image Recognition

- **Loss Curve**:



- **Confusion Matrix :**


Confusion Matrix - KNN
Class 0
Class 1
Class 2
Class 3
Class 4
True label
610
27
40
33
90
306
240
25
84
145
185
52
440
44
79
246
205
33
171
145
372
120
26
60
222
Class 0
Class 1
Class 2
Class 3
Class 4
Predicted label
600
500
400
300
200
100



- **Precision & Recall** :

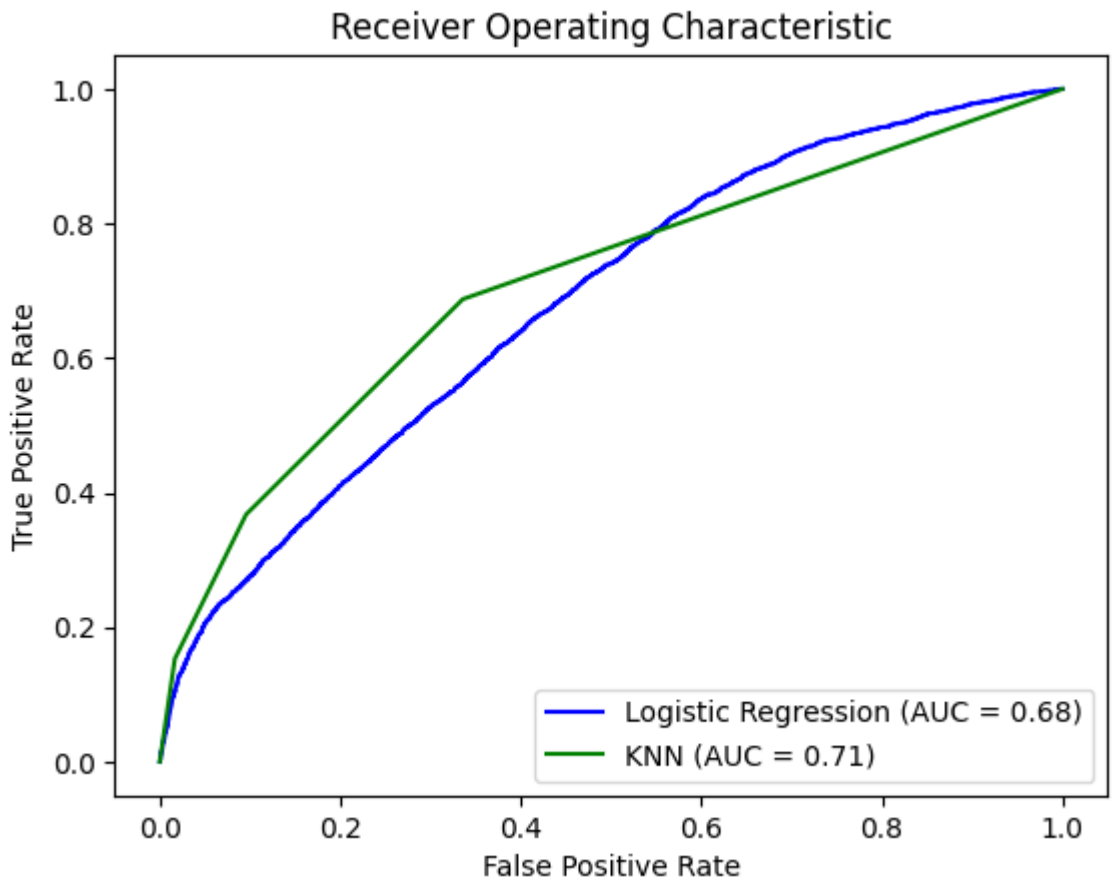
Test Accuracy: 34.75%%

Classification Report:

	precision	recall	f1-score	support
Class 0	0.32	0.70	0.44	800
Class 1	0.51	0.02	0.05	800
Class 2	0.38	0.82	0.52	800
Class 3	0.35	0.06	0.10	800
Class 4	0.32	0.14	0.19	800

accuracy			0.35	4000
macro avg	0.38	0.35	0.26	4000
weighted avg	0.38	0.35	0.26	4000

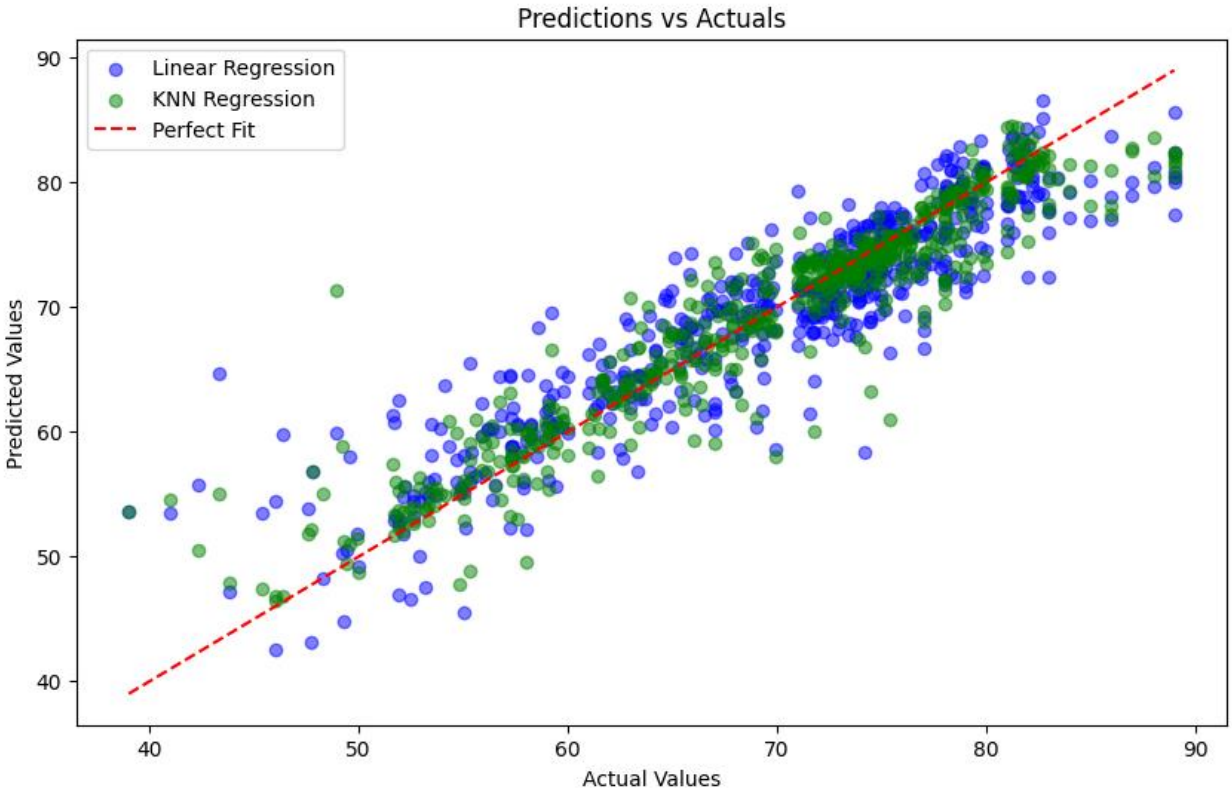
- **ROC & AUC Graph**:



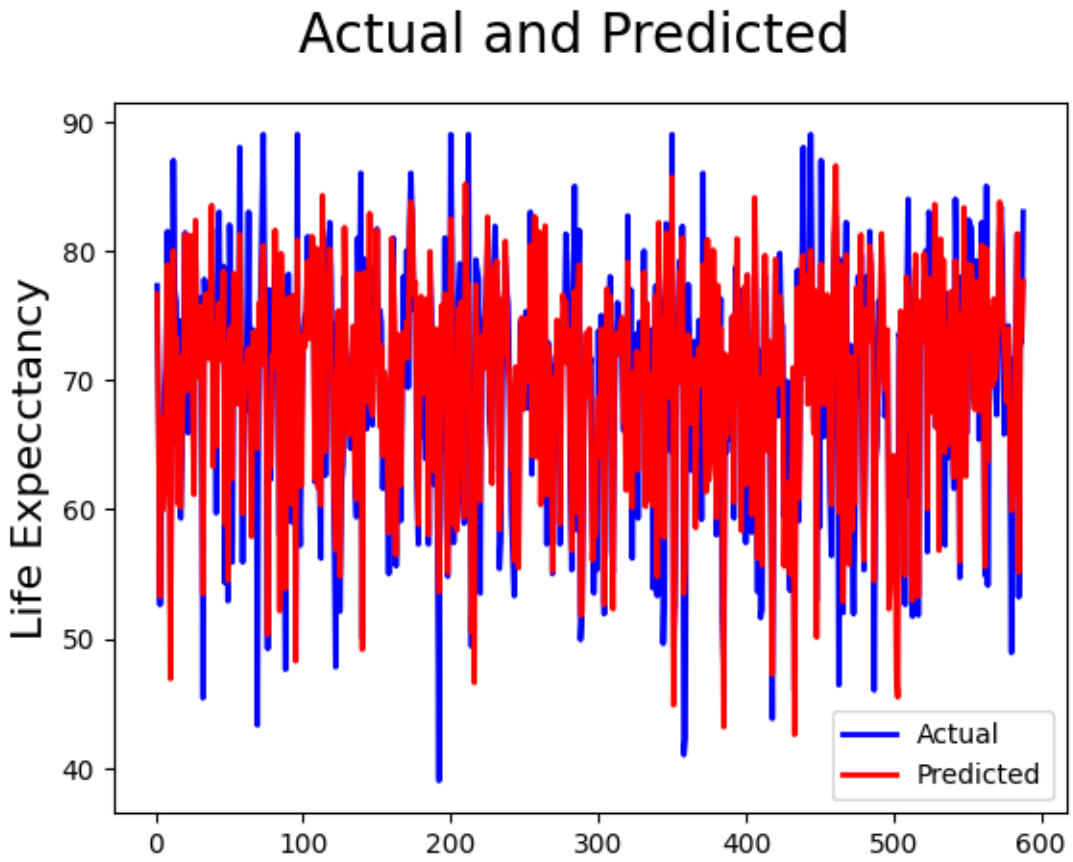
---

a. Numerical

- **Scatter Plot**:


Predictions vs Actuals
Linear Regression
KNN Regression
Perfect Fit
Predicted Values
Actual Values
90
80
70
60
50
40
40
50
60
70
80
90

- **Line Chart :**



- **Accuracy**:
- `Mean Squared Error (MSE) for Linear Regression: 16.2445499713935`

5. Final Notes

This document provides a comprehensive summary of the datasets used, algorithms applied, and their corresponding evaluation metrics. The comparison table serves as a quick reference to identify the best-performing model without navigating through multiple notebooks.