## Introduction

Linear regression establishes a relationship between a dependent variable and independent variables, often represented by a straight line. In this study, we employ two methods—Stochastic Gradient Descent (SGD) and Ordinary Least Squares (OLS)—within the context of linear regression to predict the age of abalone based on their physical measurements. Ordinary Least Squares is a common technique for estimating the coefficients of linear regression equations, with 'least squares' referring to minimizing the squared error. These coefficients, also known as parameters, describe the relationship between independent and dependent variables. On the other hand, SGD utilizes a single data point at each iteration, randomly selected, and employs the entire dataset to compute the gradient of the loss function. This gradient is then used to update the model's parameters.

## Dataset

The dataset for this study originates from the University of California, Irvine Machine Learning Repository. The same can be accessed it through this link. The original dataset comprises eight features (also known as independent variables): sex, length, diameter, height, whole weight, shucked weight, viscera weight, and shell weight. In our analysis, we excluded the categorical feature, 'sex,' and retained the remaining seven continuous features. The dependent variable (or label) is 'rings,' representing integers. The age of an abalone can be calculated using the 'rings' values. By adding 1.5 to the sum of 'rings,' we obtain the approximate number of years the abalone has lived.

The dataset consists of 4,176 instances and has no missing values. Initially, the 'StandardScaler' was applied to 'SGDRegressor' to normalize features, targeting a mean of 0 and a standard

deviation of 1. Despite this preprocessing step, it did not yield a significant difference in the output. Consequently, no preprocessing steps were incorporated during the analysis.

## Analytics

For training and testing the model, both Ordinary Least Squares (OLS) and Stochastic Gradient Descent (SGDClassifier) methods were employed. The OLS method was implemented from scratch using only NumPy and Pandas libraries, while the SGDClassifier leveraged the Scikit-Learn built-in libraries for model fitting, training, and prediction.

The dataset was partitioned into training and testing sets, constituting 80% and 20% of the data, respectively. To assess and compare the performance of both models on each set, various metrics were utilized. These metrics include mean squared error, mean absolute error, and coefficient of determination. The evaluation process involved the use of Scikit-Learn libraries to ensure consistent and reliable metric calculations. Also, seaborn library is used to visualize the actual vs predicted and residual plots.

## Results

After executing the Python code, the following results were obtained.

| **Metrics** | **Training Data** | | **Test Data** | |
|---|---|---|---|---|
| | **OLS** | **SGDRegressor** | **OLS** | **SGDRegressor** |
| Mean Squared Error | 5.12 | 5.37 | 4.08 | 4.14 |
| Mean Absolute Error | 1.65 | 1.68 | 1.54 | 1.56 |
| Coefficient of Determination | 0.54 | 0.52 | 0.44 | 0.43 |

**Parameters of the model**

The below parameter vector shows the parameters applied by both algorithms in training and test datasets. It also includes the bias term.

$w = [w_0, w_1, w_2, w_3 w_4, w_5, w_6, w_7]$

Where $w_0$ is the bias term and $w_1$ to $w_7$ are the actual parameters for respective features. Let's replace them with actual parameter values from the output of the code.

**For Ordinary Least Squares (OLS):**

$w_{OLS}$ = [3.03432495, -2.82993994, 15.38204652, 10.29982958, 9.51555164, -20.63027896, -10.15775795, 8.87471301]

**For SGDRegressor:**

$W_{SGD}$= [3.72080973, 4.35762258, 5.21591571, 3.80219415, 4.59866106, -13.47525274,

-1.47992901, 11.14085612]

The discussion of results across various metrics is presented below.

**Mean Squared Error (MSE)**

As the name implies, the mean squared error (MSE) calculates the average of the squared differences between predicted and actual outputs. It provides a consolidated measure of the model's overall average squared discrepancy between the input and output values, condensing this information into a single numerical value. A lower mean squared error indicates a smaller difference between predicted and actual outputs.

In our specific case, the disparity between the two methods is more noticeable in the training dataset compared to the test dataset. The Ordinary Least Squares (OLS) method exhibits a lower MSE in comparison to the SGDRegressor, underscoring its superior performance in minimizing the squared differences between predicted and actual outputs.

**Mean Absolute Error (MAE)**

The mean absolute error (MAE) calculates the average of the absolute differences between estimated and actual outputs. Both mean squared error (MSE) and MAE serve to quantify the disparity between predicted and actual outputs. While MSE squares the differences, MAE takes absolute values, ensuring that negative and positive differences do not cancel each other when aggregated. A lower MAE is indicative of a more accurate model.

Notably, the SGD method exhibits a slightly higher mean absolute error in both the training and test datasets compared to the OLS method. This suggests that, on average, the differences between estimated and actual outputs are marginally larger for the SGD method.

**Coefficient of Determination ($R^2$)**

The coefficient of determination serves as a metric to assess the effectiveness of a regression model relative to a baseline model. Calculated by subtracting the ratio of the regression model's mean squared error (MSE) to the baseline model's MSE from 1, $R^2$ provides insight into the proportion of variance in the dependent variable that can be predicted by the independent variable(s). $R^2$ values range from 0 to 1, with 1 signifying perfect predictability (i.e., model's MSE =0) and lower values indicate less accurate predictions. An $R^2$ of 0 indicates equal mean squared errors between the regression and baseline models.

In our specific case, the SGDRegressor exhibits a slightly lower $R^2$ in both the training and test datasets. This implies that, on average, the variation in the dependent variable explained by the independent variables is marginally lower for the SGDRegressor compared to the alternative method.

**Actual Vs Predicted Plots**

In linear regression, it is essential for the relationship between the dependent and independent variables to be linear. This assumption can be assessed by creating a scatter plot comparing the actual values against the predicted values. In the plots below, not all the points are close to the straight line, which means they don't perfectly follow the assumption of linear regression. Furthermore, the plots comparing actual versus predicted values between the two methods don't show significant differences.

**Residual Plots**

Further, to validate the model, we also assessed a fundamental assumption of linear regression by examining residual error plots for both the training and test datasets across both methods. The residual errors exhibit a normal distribution, with a mean zero for the training dataset and approximately zero for the test dataset. Similar to our analysis results, the residual plots under the OLS method reflect a slightly better performance compared to those under the SGDRegressor.

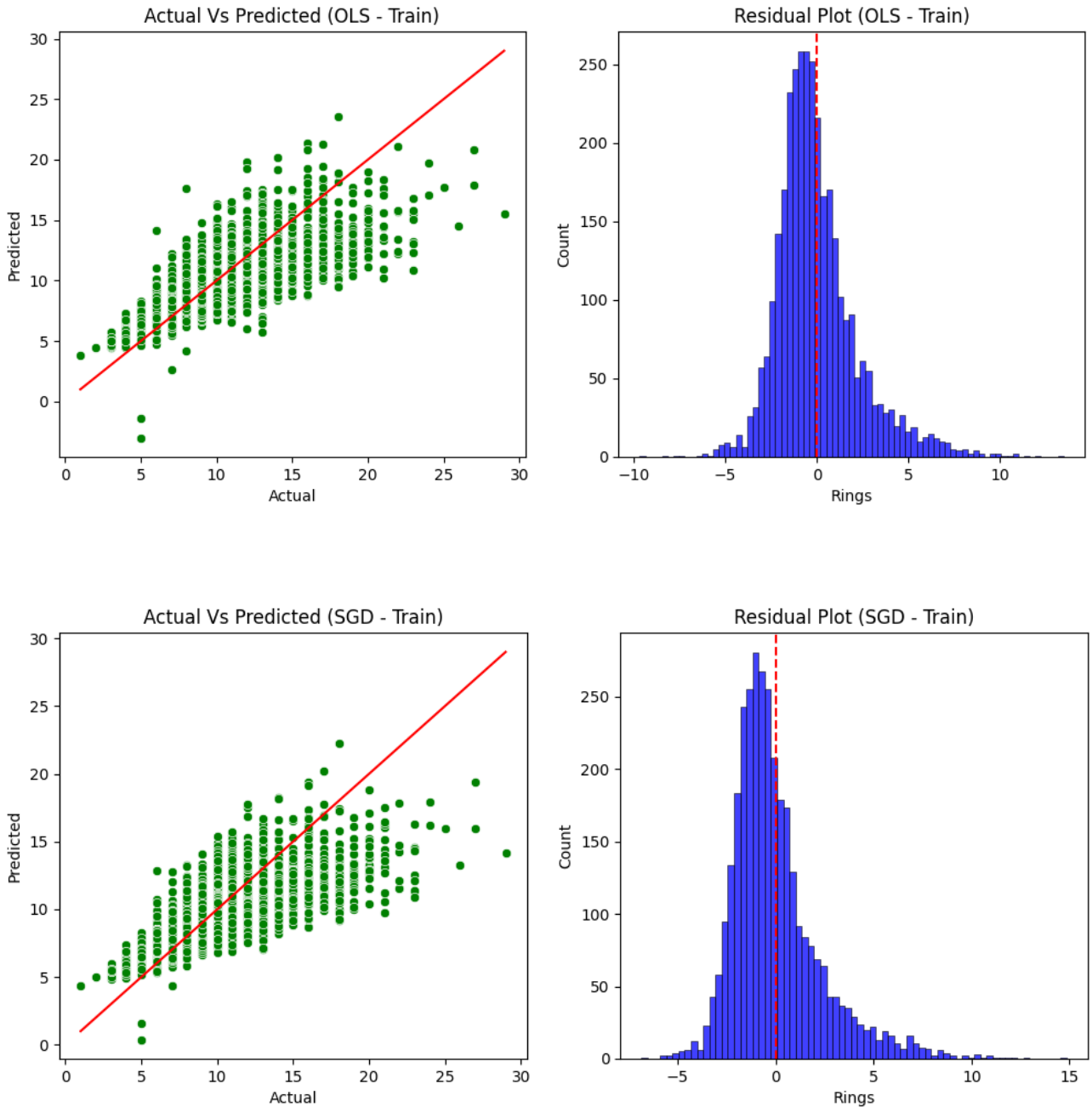


Figure 1: Plots for the training dataset

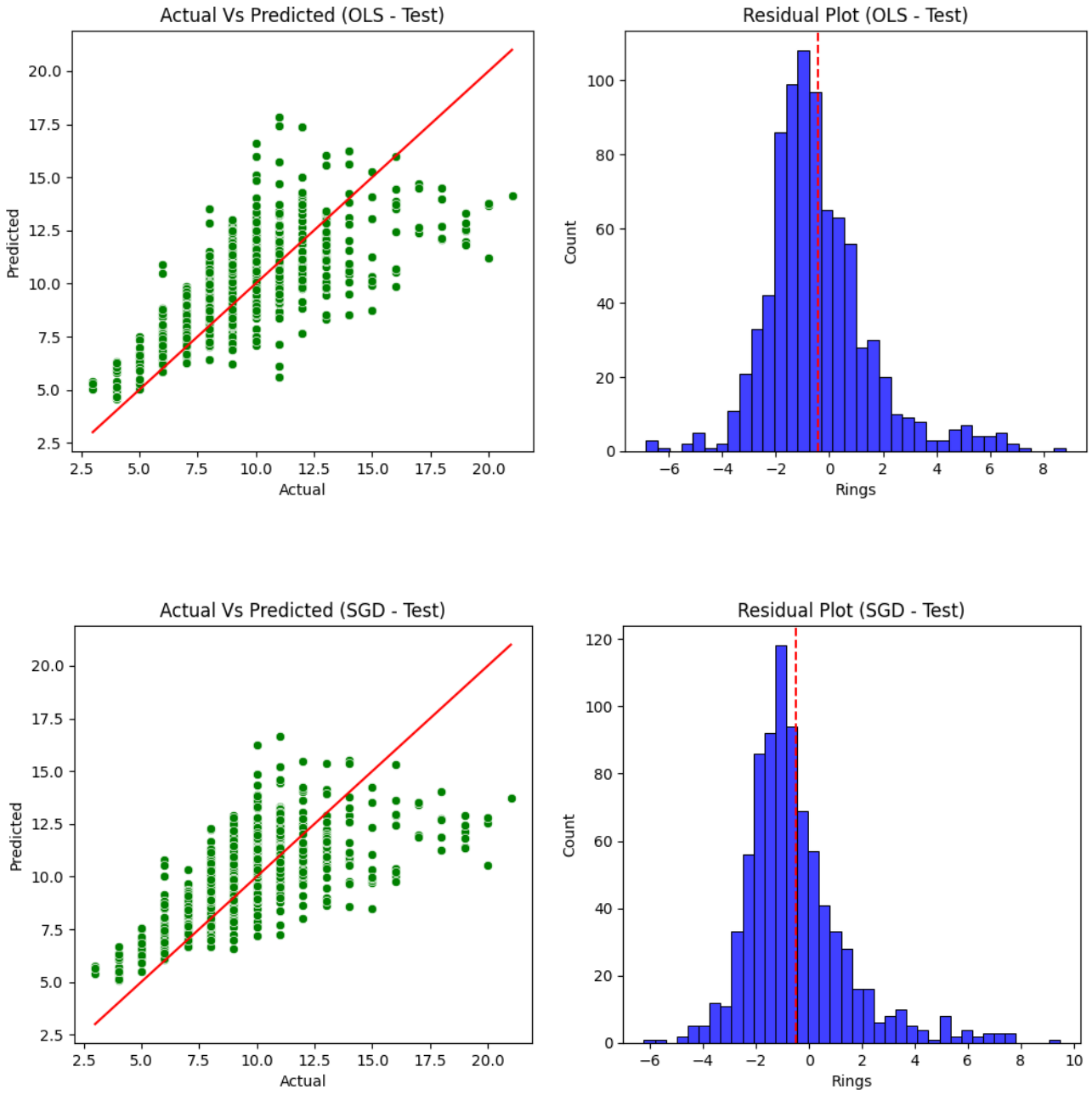


Figure 2: Plots for the test dataset