Department of Electrical and Computer Engineering

Course Project Report

ECE569A

Selected Topics Comp Engineer: Artificial Intelligence

Fish Market Analysis Model

Aniruddh Sawant (V00950050)
Sainath Padala (V00931324)

Table of Contents

List of Tables

List of Figures

Abstract

This project is based on the Fish Market Data which is available in Kaggle [1]. The dataset consists of data regarding seven different fish species. We have developed a linear regression model for estimating various features of the fish based on other measurements. Features of the fish include weight, length1 (Standard length), length2 (Fork length), length3 (Total length), height and width.

Firstly, we must select the feature that we want to predict. After selecting the feature, the data is divided into two sets: Training Dataset and Testing Dataset. Next, we use Normalization to transform the data to required scale. And then, we develop the linear regression model by evaluating the Cost function and Root Mean squared error. The two methods which we used to minimize the error are: Optimized Weights and Gradient Scale. And finally, we apply the model on the training and testing datasets to estimate the required features of the fish. By using Matplot libraries, we can also check the performance of our model by plotting the predicted values against the actual values.

In addition to the model, we used Tkinter to develop a Graphical User Interface in order to interact with the model.

1 Introduction

1.1 Background

The dataset is a record of 7 common different fish species available in the market. The dataset is available online at Kaggle [1] website. The data is taken by analyzing the amount of fish sales in the market. With this dataset, a predictive model can be implemented using machine friendly data.

1.2 Motivation

In this project, the main aspect is to predict the values for various features of fishes. Features of the fish include weight, length1 (Standard length), length2 (Fork length), length3 (Total length), height and width. So, after training the model with this dataset, we can easily predict the value of the required feature. Even though the size of the data set is small the main aim of this project is to implement the said techniques in conjunction with TKinter and design a model that can predict the selected label.
For example, fish farms can implement this model. The sellers and customers can get the measurements (weight, length etc.,) of the fish directly whenever a fish swims in front of the camera.

2 Related Work

The main objective of this project is to predict a parameter. We have developed a model based on linear regression algorithm, as this algorithm is mainly focused on predicting a value from the given input dataset.
[2] includes the study of linear regression and starts from the basics of what a model is and how it can be used to make predictions using the method under the microscope. It establishes a relation between the prediction and true label and finds the best fit by finding an optimized solution by solving the least square regression equation. One of the ways to evaluate the performance of the model is also briefly discussed.
In paper [3], the authors have performed a comparative evaluation of seven most commonly used first-order stochastic gradient-based optimization techniques. The various techniques used were the Stochastic Gradient Descent (SGD), with vanilla (vSGD), with momentum (SGDm), with momentum and nesterov (SGDm+n)), Root Mean Square Propagation (RMSProp), Adaptive Moment Estimation (Adam), Adaptive Gradient (AdaGrad), Adaptive Delta (AdaDelta), Adaptive moment estimation Extension based on infinity norm (Adamax) and Nesterov-accelerated Adaptive Moment Estimation (Nadam).
Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. Various Gradient Descent algorithms are discussed in [4]. The advantages of one over another and which conclusions on the importance of each algorithm is explained.
Normalization of feature vectors of datasets is widely used in several fields of data mining, in particular in cluster analysis, where it is used to prevent features with large numerical values from dominating in distance-based objective functions. The study in [5] covers various normalization techniques.

3 Problem Formulation

The data is taken from the Fish Market data set which is available in Kaggle [1]. This data set consists of data about seven different fish species. This dataset is a record of seven columns which define the measurements of the fish. These measurements can be interpreted as different features of the fish. For predicting the features, we have developed the model based on linear regression algorithm.
Our main goal in this project is to predict one of these features based on the data which we have. The feature that we want to predict can be taken as an input from the user and then we have to develop a model based on that. In order to develop a successful model, we must train the dataset and then test the performance of the model. The performance of the model can be judged by calculating the Root Mean Squared Error (RMSE). For better performance the RMSE should be as low as possible.
To implement this project, first we need to select the feature that we want to predict. Then we must split the dataset into two sets: Training and Testing respectively. After dividing the dataset, we have to apply normalization so as to ensure that all the features are on the same scale. After splitting the dataset, we train the model using the Training dataset and then in order to validate it we test the model against the Testing dataset. By doing this, we can get the value of MSE and we have to ensure that this value is as low as possible. Alternatively, we can also check the values of our prediction against actual values using Matplot libraries.

4 Methodology

4.1 Data Analysis

As discussed earlier, the dataset consists of seven columns and these columns can be interpreted using the below figures and table:

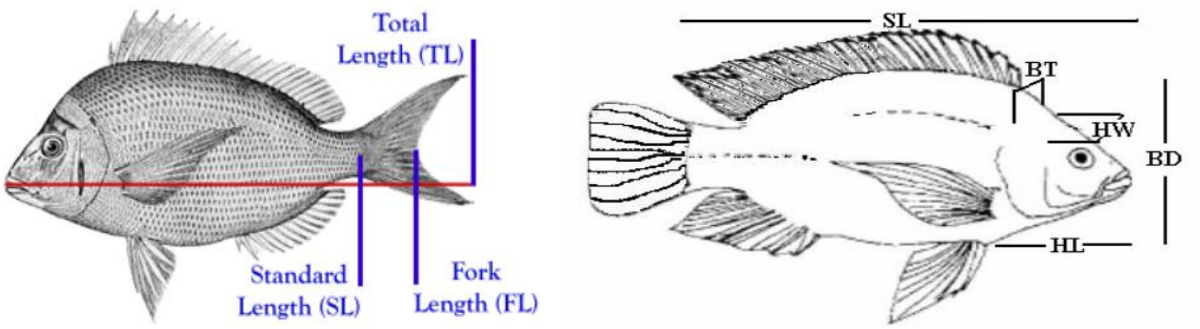


Figure 1: Various measurements of Fish

Column Name	Description	Unit
Species	Name of the Fish	
Weight	Weight of the Fish	Grams
Length1	Standard Length (SL)	Centimeters
Length2	Fork Length (FL)	Centimeters
Length3	Total Length (TL)	Centimeters
Height	Body Thickness (BT)	Centimeters
Width	Body Depth (BD)	Centimeters

Table 1: Features of Dataset

In the above table, the dependent variable is the feature we selected initially at the start of the model. For example, if we select ‘Weight’ as the label then ‘Weight’ is our dependent variable and the independent variables will be ‘Length1’, ‘Length2’, ‘Length3’, ‘Height’ and ‘Width’.

By analyzing the data, the most suitable approach to develop the model is Supervised Learning. In Supervised learning we build a model from a labeled training set. In the above-mentioned example, we know the value of the dependent variable which is weight and thus we can validate the performance of the learning algorithm. For our model, we used a linear regression algorithm. It is one of the machine learning algorithms based on supervised learning.

4.2 Linear Regression Model

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor).
The main purpose of regression is to examine if the independent variables are successful in predicting the outcome variable and which independent variables are significant predictors of the outcome.
In this project, we have developed a linear regression model which can predict the value of the selected feature (dependent variable) using the measurements of fish (independent variable).

4.3 Project Pipeline:

The below figure showcases the flow of developing the model.

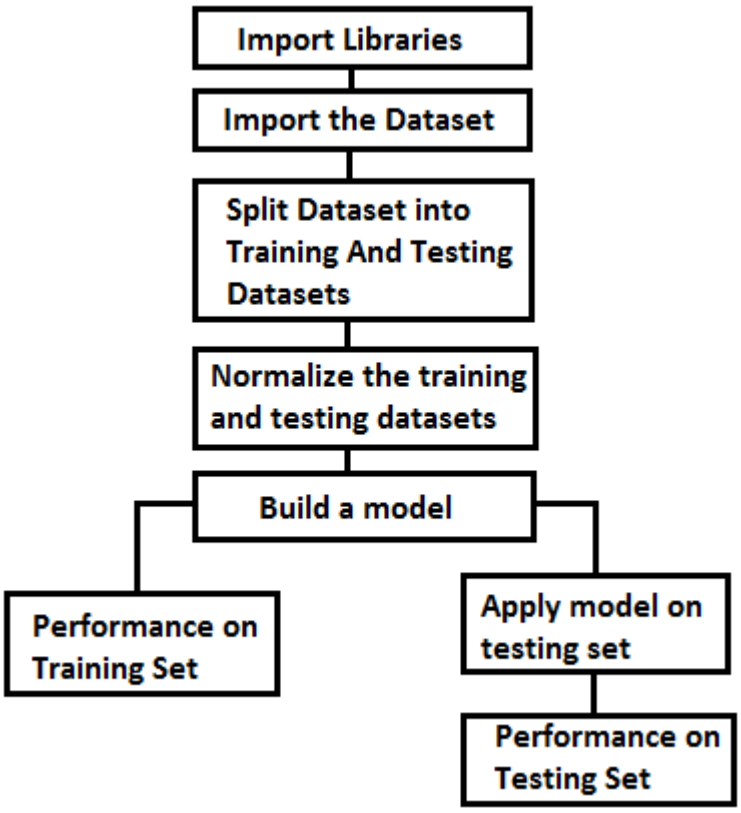


Figure 2: Flowchart of the project

The Programming language used is Python. We have developed the model without using any existing machine learning libraries. Lastly, we created a GUI interface using Tkinter in order to interact with the model.

4.3.1 Importing Relevant Libraries and Data Set

The Libraries which are used in developing this model are:

- Pandas: used for importing the dataset in order to develop model.
- NumPy: used for supporting arrays and high-level mathematical functions.
- Matplotlib: used for visualization and plotting of data.
- Tkinter: used for developing a GUI interface.
- OS: used to interact with operating system.
- SYS: is responsible for interaction between program and python interpreter.

Note: In our project, OS and SYS modules are used for setting up the RESET feature in the GUI interface.

4.3.2 Splitting the Data Set

To train any machine learning model, we must split the entire data. This allows us to tune various parameters of the algorithm without making judgements that specifically conform to training data.
Training Set is used to train the learning algorithm and the Testing Set is used to test the remaining data by checking how well the algorithm is evaluating with unknown data. In our model, we split the entire dataset into 2 parts: Training and Testing.

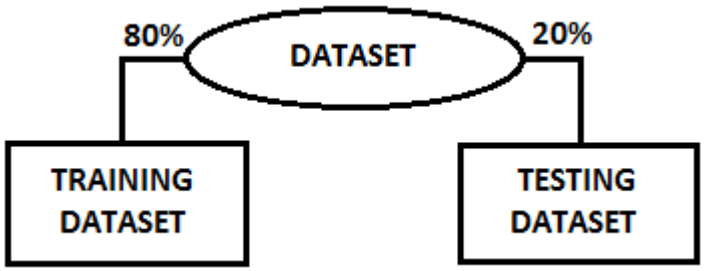


Figure 3: Splitting the Dataset

For splitting the data set according to requirement of user, we have introduced a parameter called “Split Ratio” in our model and GUI interface. If Split Ratio is selected as “0.8” then the data set is divided as shown in the above figure.

4.3.3 Normalization:

Normalization is used to improve the performance and training stability of the model. It is also necessary to speed up the learning algorithm. It transforms different attributes so that they will be on a similar scale. Few of the techniques which we used in this model are:

- Standardization
- Min-Max Scaling

4.3.3.1 Standardization

Standardization is a technique in which the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

$$X' = \frac{X - \mu}{\sigma}$$

4.3.3.2 Min-Max Scaling

Min-Max Scaling is a technique in which values are shifted and rescaled so that they end up ranging between 0 and 1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Another easy way of scaling the data is shown as follows:

$$X' = \frac{X}{X_{max}}$$

4.3.4 Building Model:

As discussed earlier, linear regression performs the task to predict a dependent variable (weight of the fish) based on given independent variables (measurements of fish). So, this technique is used to find a linear relationship between them.

The equation for this algorithm is given by:

$$y = wx + b$$

where, y is the dependent variable (predicted value)
x is the independent variable
w is the slope of the line
b is the y-intercept or bias

4.3.4.1 Cost Function

The Cost Function used in linear regression is Mean Squared Error (MSE). This Cost function is used to calculate the error difference between the predicted value and true value.

The equation for calculating error is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (wx_i + b))^2$$

where, n is the total number of observations
$\frac{1}{N}\sum_{i=1}^{n} is\ the\ mean$
y_i is the actual value of an observation and $wx_i + b$ is our prediction.

4.3.4.2 Optimized Weights

With the data as {(X, y)} where X are the samples or features and y are the labels, where size of X is NxP, N is the number of features and P is the total number of samples. We then augment the data by ones as, $X_{new} = [X\ 1]^T$ for each sample of each feature changing the dimensions to (N+1)xP. The labels in the data are arranged in a column vector as Px1.

From [write reference number], the optimal weight w and bias b for model y = wx + b are given by w* = $(X\,X^T)^{-1}$ X y and w* = $[w\ b]^T$.

4.3.4.3 Gradient Descent

In order to minimize the cost function, we have used a well-known optimization algorithm, Gradient Descent. Basically, it tries to find the local optima of the function by taking "steps" until it hopefully converges. The "step" is done by taking the derivative (the line tangent to a function) of the cost function. The slope of the tangent is the derivative at that point and gives the direction to move towards.

Step 1: Calculating the partial derivative of the above Cost function.

$$dw = \frac{\partial(MSE)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2\right)$$

$$db = \frac{\partial(MSE)}{\partial b} = \frac{\partial}{\partial b}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2\right)$$

Step 2: Now, we have to update the current values of 'w' and 'b'.

$$w = w - lr*dw$$

$$b = b - lr*db$$

where, lr is the learning rate.

These steps are repeated a number of times until an optimum value is reached. We can give number of iterations to the function and accuracy increases with each iteration.

4.3.5 Applying the model on Data Set

After implementing the model using Python, we have developed a GUI application interface using Tkinter. Using this we can interact with the model. Below screenshots show the GUI interface for both Optimized Weights and Gradient Descent algorithms.

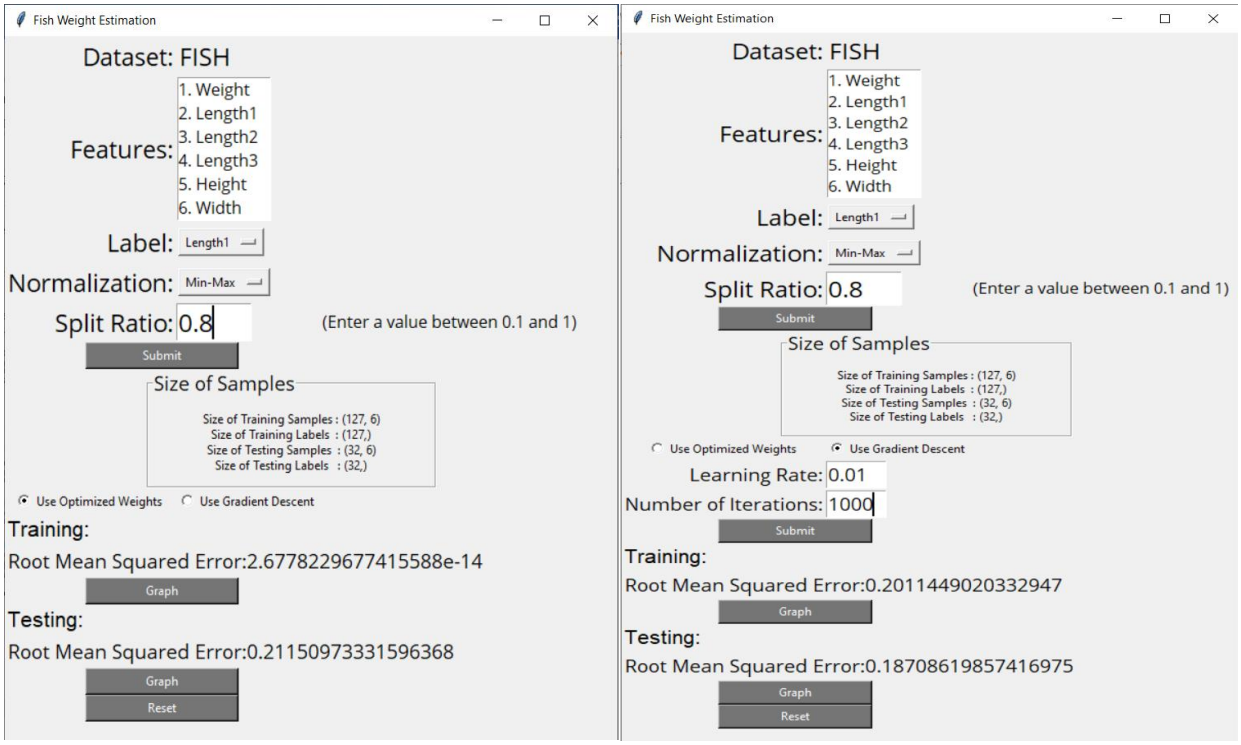


Figure 4: GUI Interface for the model

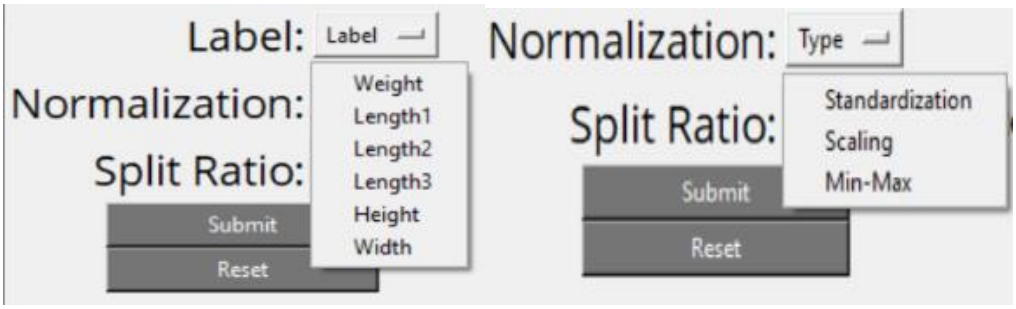


Figure 5: Types of Labels and Normalization

From Figure 5, we can see that we can choose any of the features from Label Drop-down and normalization type from Normalization Drop-down.

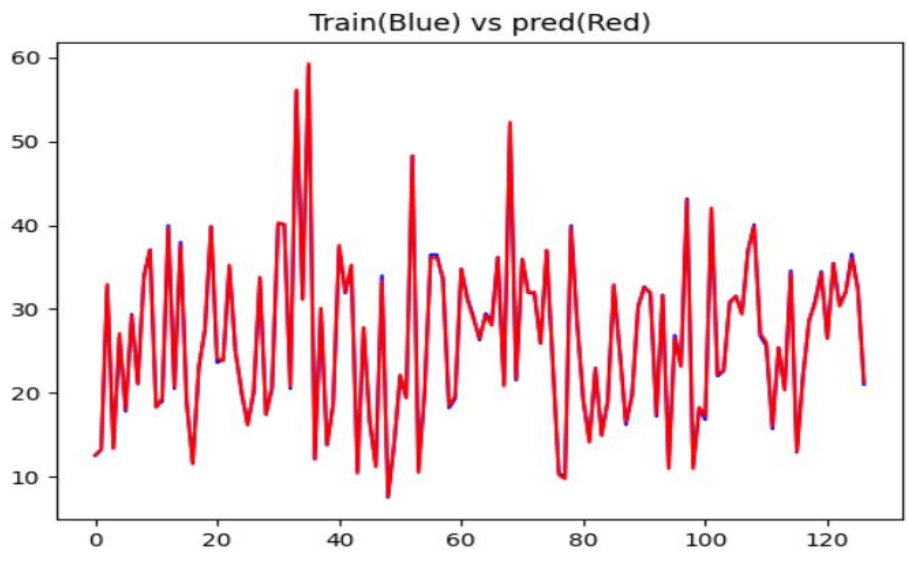
For the rest of the report, we have used below values:

Label	Length1
Normalization	Min-Max
Split Ratio	0.8

Table 2: Values for Label, Normalization, Split Ratio

4.3.5.1 Using Optimized Weights:

By selecting the optimized weights through GUI, we can use this algorithm to calculate the Root Mean Squared Error and plot the corresponding regression graph.
From *Figure 4*, we get those RMSE values when we select the values as shown in *Table 2*. After calculating the error value, we visualize the corresponding plots by using the Matplotlib library present in python. The predictions which were generated are as shown below:

Data Set	**Prediction**
Training Set	

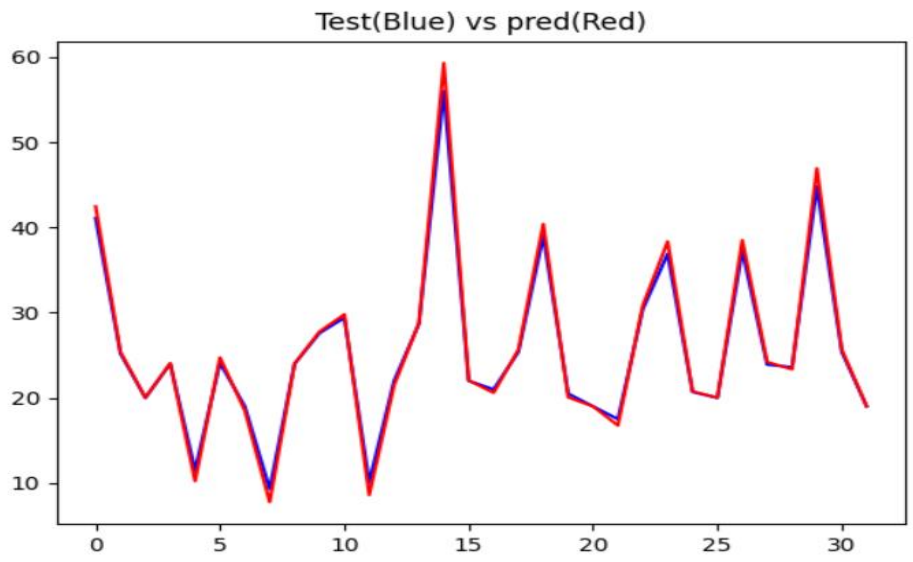
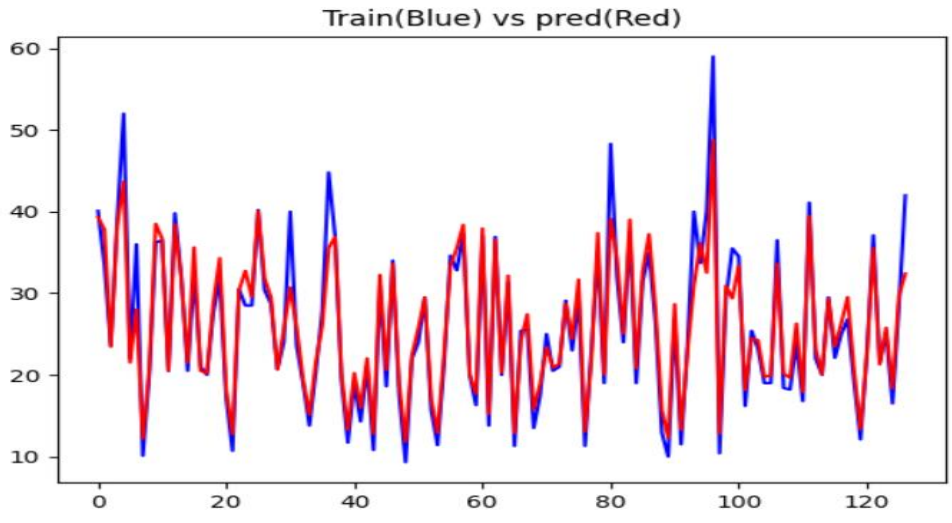
Testing Set	
Blue: This line indicates the actual value from the data set. Red: This line indicates the value which we predicted using the optimized weights method.	

Table 3: Plots when Optimized weights is selected

Note: We can change the Label, Normalization method and Split Ratio depending on our requirement at the start of the model. For each run, the splitting will be different and random.

4.3.5.2 Gradient Descent

Similar to the above method, this method can also be selected from the GUI interface.
From *Figure 4*, we can see that there are two additional input fields (Learning Rate, Number of Iterations) in addition to the remaining fields (Label, Normalization Type, Split Ratio). For the below table, we have selected the same values as shown in *Table 2.* But in addition to that we have given learning rate LR = 0.01 and Number of Iterations = 1000.

Dataset	**Prediction**
Training Set	

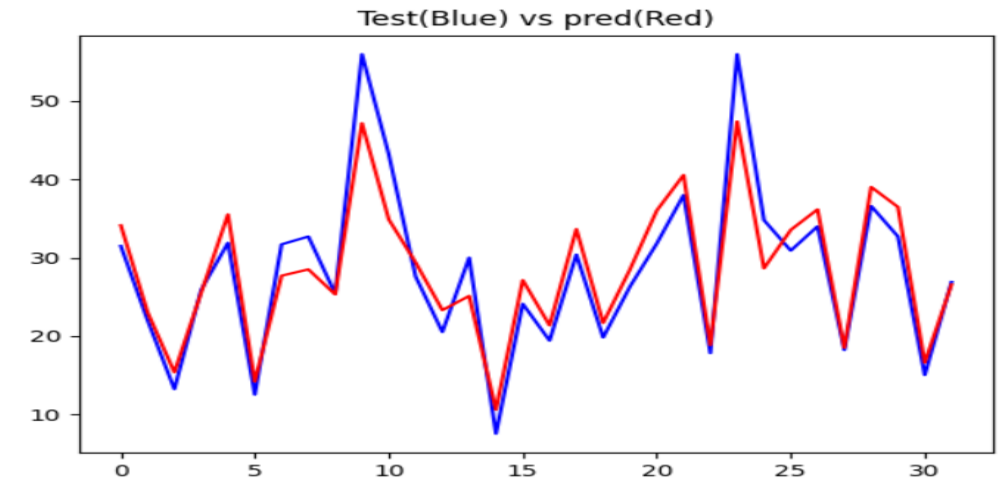
Testing Set	
Blue: This line indicates the actual value from the data set. Red: This line indicates the value which we predicted using the optimized weights method.	

Table 4: Plots when Gradient Descent is selected

4.3.6 GUI using Tkinter

We have developed the Tkinter GUI application by importing an inbuilt library of Python. From *Figure 4* and *Figure 5*, we can see that the interface uses multiple functionalities like Labels, Drop-down, Entry Box (Input field), Buttons, Radio button, Text boxes and Reset Button. We have implemented these functionalities in parallel with the code of the model.

Note: The RESET button restarts the whole program from start by closing the current window and starting a new window.

5 Results and Discussions

The performance of the model is calculated using the Root Mean Squared Error (RMSE). Below are the RMSE values which we got as a result of running the model with settings mentioned in 4.3.5

Method	Root Mean Squared Error	
	Training	Testing
Optimized Weights	2.677×10^{-14}	0.21
Gradient Descent	0.201	0.187

Table 5: Root Mean Squared Error values when Length1 is selected

From the above table, we can clearly say that the Root Mean Squared Error (RMSE) is very less for both training and testing datasets. Thus, we can say that the performance of our model is good.

Furthermore, from *Table 3* and *Table 4*, we can see that the predictions (RED) from plots are very close to that of the actual values (BLUE).

6 Conclusion

We have successfully achieved our main goal, that is to predict the value of the feature which was selected at the start of the model. In the above figures and tables, the predicted feature is “Length1”, however we can predict other features also using the same method. We have also checked the performance of the model by checking the error value and predicted plots.

7 References

[1] https://www.kaggle.com/aungpyaeap/fish-market
[2] Statistics for Research Projects
[3] A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks
[4] An overview of gradient descent optimization algorithms
[5] Statistical approach to normalization of feature vectors and clustering of mixed datasets
[6] A Multiple Linear Regression Approach for estimating the Market Value of Football Players in Forward Position - Yunus Koloğlu, Hasan Birinci, Sevde Ilgaz Kanalmaz, Burhan Özyılmaz.
[7] Machine Learning Benchmarks and Random Forest Regression - Mark R. Segal
[8] A Regression Equation Model for Height and Weight Prediction - Xu ZHANG, Chao ZHAO, Hai-tao WANG, Fang ZHANG,,Jing ZHAO2 and Gang WU