# **Introduction**

We perform this work's regression and classification task by manually developing machine learning models. For the regression task, a dataset of samples N=50 was generated manually and used to train and evaluate the developed model. For the classification task, the built-in iris dataset was used to train and evaluate the developed classifier.

# **Dataset**

For the regression task, the training set and the testing set were generated using the random number function and eq 1. For the training and testing set, the single feature x was used and generated from the random number function between the range of -5 and 5. After getting the random number as x, the target variable y was calculated using eq 1. The 30 samples were generated for the training set and 20 samples for the testing set. The scatter plot on the training set is presented in Figure 1 to analyze the distribution of the dataset.

Eq. 1

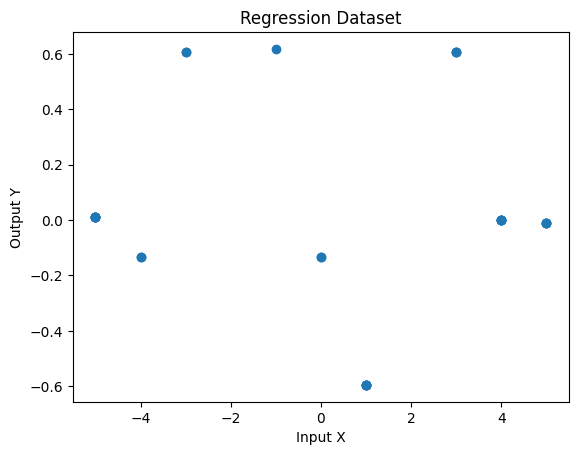


Figure 1: Scatter Plot of Dataset for Regression with Degree 2.

The built-in iris dataset was used to perform the classification task. The iris dataset was loaded from the dataset’s module of scikit-learn library. The loaded dataset was based on the 150 samples and each sample has the 4 feature values and corresponding labels. Basically, the iris dataset contained samples of three types of flowers. The feature columns represent the sepal length (cm), sepal width (cm), petal length (cm), and petal width (cm) of each flower. The label or target column of the dataset represents the class or types of the flower. The dataset contains samples of the setosa, [Versicolor](https://en.wikipedia.org/wiki/Iris_versicolor), and Virginia classes of the flower represented by the 0, 1 and 2 respectively. The class distribution of the iris dataset is presented in Figure 2. The class distribution analysis reflects the number of samples for each class in the dataset.

# **Regression**

**Model Development and Training**

Chart, bar chart

Description automatically generatedThe regression model was developed for the mapping of input variables x1 to xn on the target variable y1 to yn. The regression model was developed by initializing the random weights equal to the length of features. Further, the feature values were multiplied by the random weights and updated gradually.

The training set was used for the learning of the model. For each sample, the feature value was multiplied by the corresponding weight. The randomly generated weights were updated after multiplying the feature values. Lastly, the updated weights were used for the evaluation of the trained regression model.

**Model Evaluation Measures**

Figure 2: Class Distribution of Iris Dataset for Classification.

By following the training of the regression model, numerous evaluation measures were used for the evaluation of the model. The mean squared error (MSE), root mean square error (RMSE), and mean absolute error (MAE). Lastly, the generated test samples (N=20) by eq 1 were used to evaluate the model. The equations of selected evaluation measures that are described below were used to calculate the results.

**Mean Absolute Error:** Mean Absolute Error (MAE) is a commonly used metric for evaluating the performance of a regression model. It is used to measure the difference between the predicted values and the true values. The MAE is calculated as the average of the absolute differences between the predicted values and the true values. Mathematically, it is represented in Eq 2. where yi is the true value, ŷi is the predicted value, and n is the total number of samples.

**Mean Squared Error:** The MSE is calculated as the average of the squared differences between the predicted values and the true values. Mathematically, it is represented in Eq 3. where yi is the true value, ŷi is the predicted value, and n is the total number of samples.

Eq 3

**Root Mean Squared Error:** RMSE is the square root of the Mean Squared Error (MSE) which is calculated as the average of the squared differences between the predicted values and the true values. Mathematically, it is represented in Eq 4. Where yi is the true value, ŷi is the predicted value, and n is the total number of samples.

Eq 4

**Model Results**

The developed model was trained with the different variants of the dataset. The training samples(N=30) were generated with the polynomial degree of 2, 3, 4 and 5 in eq 1. The training samples were used for the training of the model. The test samples (N=20) with the same degree of polynomial were generated for the evaluation of the model. The extracted result on the test samples using the eq 2-4. The extracted results are presented in Table 1 and shown in Figure 3.

Table 1 and Figure 3 presented the testing set generated with the degree 2 was outer performed on the proposed model compared to the other datasets. The increase in the polynomial degree decreased the value of the target variable y. The values too much small or much large are probably not learned properly by the proposed model. Hence the proposed model showed the minimum error with the degree of 2 and the maximum error with the degree of 5.

Table 1: Evaluation Measures for Regression Task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Degree2** | **Degree3** | **Degree4** | **Degree5** |
| **MAE** | 0.352413 | 0.429138 | 0.389977 | 0.486385 |
| **MSE** | 0.225695 | 0.29372 | 0.282977 | 0.358268 |
| **RMSE** | 0.475074 | 0.541959 | 0.531956 | 0.598555 |

Chart, bar chart

Description automatically generated

Figure 3: Comparison Results of Trained Models.

# **Classification**

**Model Development and Training**

For the classification of the iris dataset, the model was proposed with only a single layer. On the single layer, the neurons were set equal to the length of features in the iris dataset. Firstly, the random weights were initialized and then updated gradually by multiplying them with the feature values of each sample. After the multiplication of features with weights, the loss was calculated and update the weights accordingly. After the development of the model, the dataset was split into two different subsets labelled as the training set and testing set. For the splitting of the dataset, the train test split function of scikit-learn library was used. Further, the training samples (N=120) were used for the learning of the proposed classification model.

**Model Evaluation Measures**

By following the training of the proposed classifier model, the test samples (N=30) of the iris dataset were used for the evaluation of the model. Numerous evaluation measures including accuracy, precision, recall and f1-score were used for the evaluation of the model. The selected evaluation measures were calculated using the below formula of corresponding evaluation measure:

Eq. 5

Eq. 6

Eq. 7

Eq. 8

**Model Results**

The proposed model was tuned with the learning rate as a hyperparameter. The learning rate with 0.1, 0.01, 0.001 and 0.0001 was used for the training of the model. The results of the proposed classifier are presented in Table 2 and shown in Figure 4. The Table 2 and Figure 4 showed that the proposed model outer performs with the learning rate of 0.0001 compared to the other trained model. The below table showed the maximum accuracy, precision, recall and f1-score with the 0.0001 learning rate.

Table 2: Results of Evaluation Measures for Classification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metrics | 0.1 | 0.01 | 0.001 | 0.0001 |
| accuracy | 0.9 | 0.633333 | 0.9 | 1 |
| precision | 0.907407 | 0.45 | 0.928571 | 1 |
| recall | 0.909091 | 0.666667 | 0.909091 | 1 |
| f1-score | 0.903581 | 0.532855 | 0.907368 | 1 |

Chart, bar chart

Description automatically generated

Figure 4: Comparison Results of Trained Models.