**Assignment 1**

**Introduction:**

This is an assignment on implementation of linear models for Regression and classification. We implement models without using sklearn, thereby using our own models to gain deeper understanding of the concepts.

**Assignment overview:**

This assignment consists of two parts:

1. **Implementation of the linear regression class:** In this section, the fit, predict and score methods are implemented. We perform regression tasks on single output and multiple outputs.
2. **Classification:** In this section, we implement the logistic regression class is implemented in the same way as linear regression class.

**Dataset:**

We use the iris dataset from sklearn library. It is a popular dataset in Machine learning for beginners. The dataset is split into training and test sets. 90% of data is for training the models and the rest is for test set.

**Part 1: Linear regression**

In this section, we implement the linear regression class by implementing the **fit, predict and score methods**. The fit method uses gradient descent to optimize the model parameters using mean squared error as the loss function. By default, model will train for 100 iterations, or the patience reaches 3 which acts as our stopping rule. The predicted values in the predict function is given by y = XW + b, where X is the input, W is the weight and b is for bias. The score method returns the loss function which is the mean squared error.

A math equation with numbers and symbols

Description automatically generated

After training, the model parameters are saved into respective files. This was done by implementing the **save and load functions**. Later the model parameters can be imported into the testing script and can be tested on the test set. This file is named as LinearRegression.py inside the LinearRegression folder.



There are four different combinations of input and output features that are used to train 4 different models and the models’ parameters are saved in their files. The training files are saved as *SLR[model\_no]\_Train.py*. The model parmaeters are saved in *.npz* files. The model testing is performed in the files named as *SLR[model\_no]\_Eval.py.*

**Model 1:**

In this model, we predict sepal length using sepal width. The two plots below represent the model without regularization and Regularized model. The difference between parameters were displayed in the output.

**A graph with blue lines

Description automatically generatedA graph with blue lines

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**Model 2:**

In this model, we predict petal length using sepal width. The two plots below represent the model without regularization and Regularized model. The difference between parameters were displayed in the output.

**A computer screen shot of a program

Description automatically generated**

**A graph with blue lines

Description automatically generatedA graph with a line graph

Description automatically generated**

**Model 3:**

In this model, we predict petal length using petal width. The two plots below represent the model without regularization and Regularized model. The difference between parameters were displayed in the output.

**A graph with a line graph

Description automatically generatedA graph with a line graph

Description automatically generated**

**A computer screen shot of a program code

Description automatically generated**

**Model 4:**

In this model, we predict petal width using sepal length. The two plots below represent the model without regularization and Regularized model. The difference between parameters were displayed in the output.

**A graph with blue lines

Description automatically generatedA graph with blue lines

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

In the above models, model 1 performs better than all other models. The models 2 performs the worst of all models. The parameters for each model and there is displayed in the output.

**Model with multiple outputs:**

In this model, we predict sepal length using sepal width. The two plots below represent the model without regularization and Regularized model. The difference between parameters were displayed in the output.

**A computer screen with white text

Description automatically generated**

**A graph with a line graph

Description automatically generated**

**A graph with a line graph

Description automatically generated**

The linear regression performs well for 2 inputs and 2 outputs. It has MSE of 0.27 which is pretty good. The model performs really well in the test set. The Model parameters and their differences for both regularized and non-regularized are displayed in the output.

**Part 2: Classification**

In this section, we perform classification using logistic regression. The fit, predict methods are implemented the same way as linear regression. We use sigmoid function in this method. There are 3 models, each trained on a different variant in the dataset. The testing is performed in eval\_classifier.py files.

**Model 1:**

In this model, we classify petal length and width. The plot is saved in the current directory. The accuracy of the model is displayed in the output.

**A diagram of a blue and orange chart

Description automatically generated**

****

The accuracy of the model is 0.6 which acceptable which means the model is not performing well in classification of the data.

**Model 2:**

In this model, we classify petal length and width. The plot is saved in the current directory. The accuracy of the model is displayed in the output.

**A number on a black background

Description automatically generated**

**A diagram of a blue and orange triangle with green and orange dots

Description automatically generated with medium confidence**

This model has accuracy of 0.8 which is a pretty good score. This means that the model is able to classify the data.

**Model 3:**

In this model, we classify all features in the dataset. The plot for model 3 cannot be generated due to its complexity in visualizing classifying all the features in the dataset. The accuracy of the model is displayed in the output. The model has an accuracy of 0.63 which is not bad given the complexity of the model.



**Conclusion:**

In this assignment, we have implemented the linear regression and logistic regression class. We have used MSE for linear regression and Sigmoid for Logistic regression as the cost functions. The models were made based on various features and combinations. Regularization is applied to the models and the differences in the parameters were compared. The Model1 in linear regression performs well compared to others. The model with multiple outputs performs well with a MSE of 0.27. In logistic regression section we have classified 3 variants of models. The petal length/width gave an accuracy of 0.63 whereas the model 2 gave 0.8. Both their plots were visualized using mlxtend library as mentioned. For model3 the plot cannot be made because of model’s complexity. This concludes the assignment.