

# Report - *Linear Regression*

Assignment 3 · Dinesh Jaykumar Kodwani

**Part 2:** *Make predictions by using your implementation*

## Data collection/generation:

Data is *artificially generated* using a renowned machine learning open-source library called scikit-learn.

The generated dataset has the following dimensions:

- Samples/Data-instances( $n$ ) = 200
- Features ( $d$ ) = 4

*Random-state is set to 1, for reproducibility.*

The generated data consists of:

- $X$  (input): Matrix of 200 data samples, each with 4 features.
- $Y$  (output): Matrix of the target variables corresponding to  $X$ .

## Data Splitting

Dataset is split 50-50 percent into train and test. Data instances are alternatively assigned to test and train datasets.

## Hyperparameters

To train the model, we change alpha (learning rate) and epochs (1 epoch means the model computes going through the whole dataset 1 time). Various alpha and epoch values are tested with a goal to have loss lower than  $(1e-2)$ . (see *Table 1*)

## Model Training and Loss Calculation

Using the linear regression function from part 1, the model is trained on this dataset's training data. This results in a vector of weights.

Using these weights and the training data, target labels are predicted.

Further, loss is calculated for training data and testing data by calculating the Mean Squared Error between predicted and actual target labels.

In order to minimize the loss/cost function, alpha and epoch values are altered following a gradient descent approach. (see Table 1)

alpha=0.0001, n_epoch=100	<pre> trained_weights (w) : [0.1888995  0.46706119 0.4701277  0.73917181] Training Loss: 5803.508784846219 Training Loss: 6073.423789031006 </pre>
alpha=0.001, n_epoch=100	<pre> trained_weights (w) : [1.80848521 4.47556644 4.603706  7.04561834] Training Loss: 4963.9620199457195 Testing Loss: 5232.868297430126 </pre>
alpha=0.001, n_epoch=1000	<pre> trained_weights (w) : [12.51533654 30.05272454 36.51306709 45.94906067] Training Loss: 1102.0213446265777 Testing Loss: 1223.4043371666112 </pre>
alpha=0.005, n_epoch=1000	<pre> trained_weights (w) : [20.88862455 47.03051258 72.99325436 74.89838326] Training Loss: 2.382755103115634 Testing Loss: 2.8762809859106877 </pre>
alpha=0.01, n_epoch=1000	<pre> trained_weights (w) : [21.078077  46.97589201 75.10722798 76.22214929] Training Loss: 0.002183938008619034 Testing Loss: 0.0028721752802048743 Test loss is lower than 1e-2 </pre>

**Table 1:** Changes in loss with respect to change in step-size/learning rate(alpha) and number of epochs(n\_epochs)

## Conclusion

It can be observed that the loss goes below ( $1e-2$ ) at the learning rate/step size (alpha) of 0.01 and 1000 epochs (n\_epoch).