**Logo, company name

Description automatically generated**

**Neural Network and Deep Learning**

**Assignment 2**

**Submitted To:**

**Dr.Alioune Ngom**

**Submitted By:**

**Aayushi Navinchandra Patel (Student ID:110087817)**

**Aaditya Pradipbhai Parekh (Student ID:- 110084734)**

**Dhruvkumar Arvind Patel (Student ID:- 110055817)**

**Mrinal Walia (Student ID:- 110066886)**

**Date: 31-05-2022**

**GitHub Link: https://github.com/abhiwalia15/COMP8610\_Assignment\_2**

**Question 1: In this question you are to create some simulated data sets and then use the Adaline neuron and the Sigmoid to perform some prediction. Use whatever programming language you want to use.**

**Generate 5000 synthetic data points (*x*, *y*) as follows:**

* + Using the rnorm() function in R (or equivalent in Matlab or Python or etc), create a vector, *x*, containing 5000 observations drawn from a Gaussian distribution *N*(0, 1) [i.e., a normal distribution with mean 0 and variance 1]. This vector *x* represents your set of inputs *x*.

* + Using the rnorm() function in R (or equivalent in Matlab or Python or etc), create a vector, *eps*, containing 5000 observation drawn from a *N*(0, 0.25) distribution; i.e., a normal distribution with mean 0 and variance 0.25.

* + Using vectors *x* and *eps*, generate a vector *y* according to the model

*y = -*1*+*0.5*x –*2*x*2*+*0.3*x*3*+ eps*.

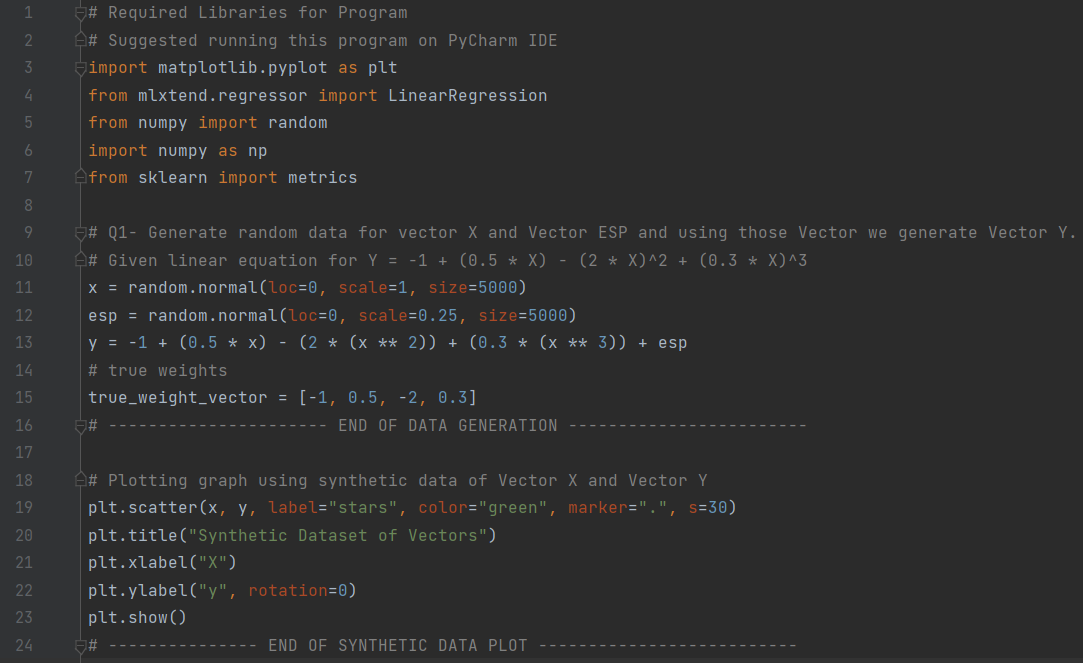
Your 5000 data-points (*x*, *y*) are generated upon completion of this Part-c. Note that the true function is a cubic function with true weight vector being *wtrue =*(*-*1, *+*0.5, *-*2, *+*0.3).

* + Implement the Adaline and Sigmoid neuron learning algorithms using (i) batch gradient descent [BGD] and (ii) stochastic gradient descent [SGD]. Using a cross-validation method of your choice (LOOCV or 10-fold-cv), test and compare their regression performances over the synthetic dataset created above. The initializations, the learning rate, the size of test set and training set, and the stopping criterion, and etc are left for you to explore. Think about the reasons why you use a particular strategy. Use your creativity and perform whatever experiments you want to test, and then tell me whatever story your experiments told you.

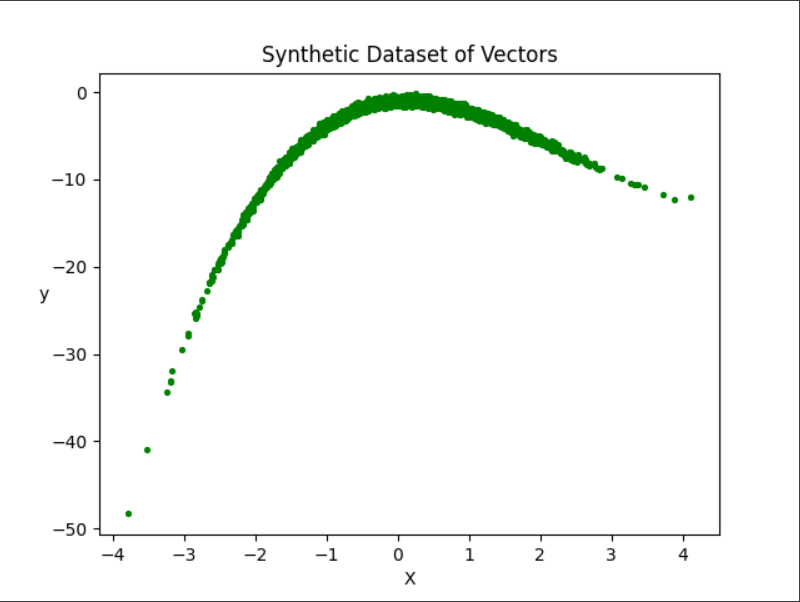
**Solution:**

The below code solves the given question step by step with all required explanations:

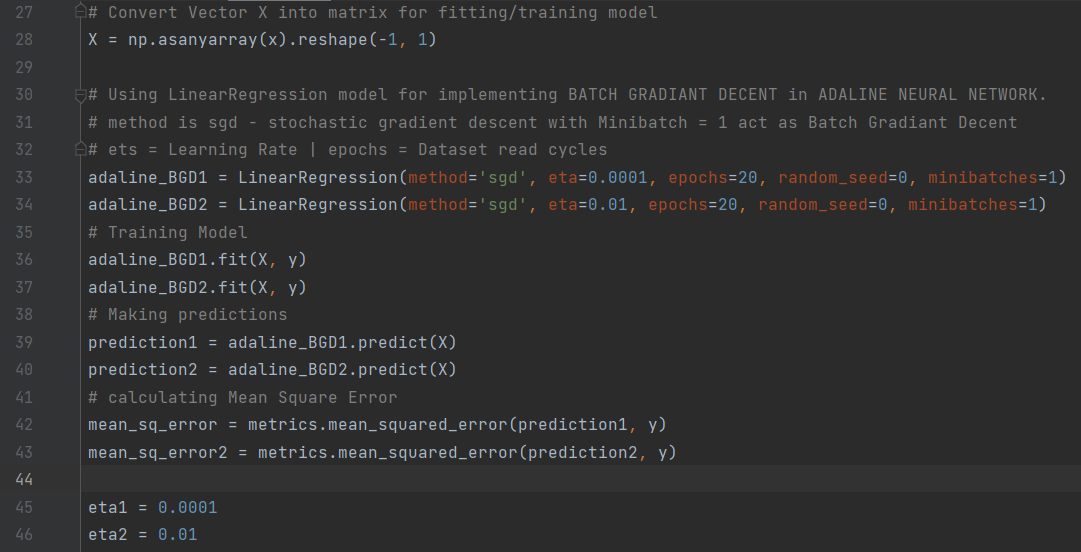
1. Using function rnorm(), we first generate vector x and vector eps. Then we calculate vector y using vector x and vector eps.



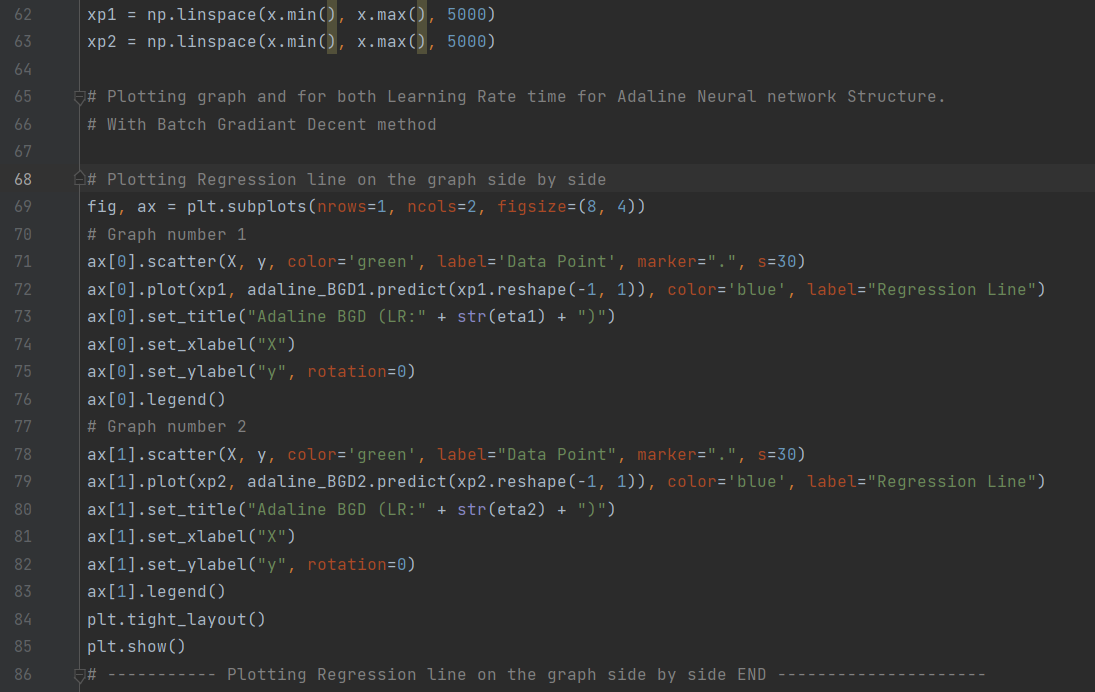
**Output:** The output plots a gaussian distribution of x and y points.



1. In this part we implement Adaline Neuron Learning Algorithm using batch gradient descent (BGD) formula. The code for that can be checked below:

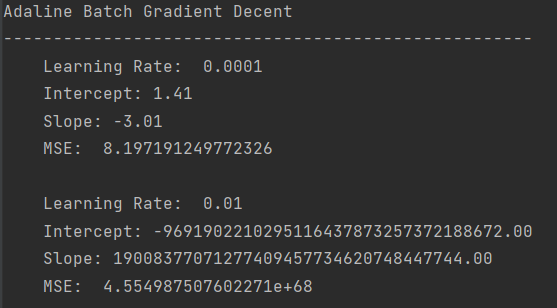


Then we run and test the algorithm on our synthetic dataset we have created.

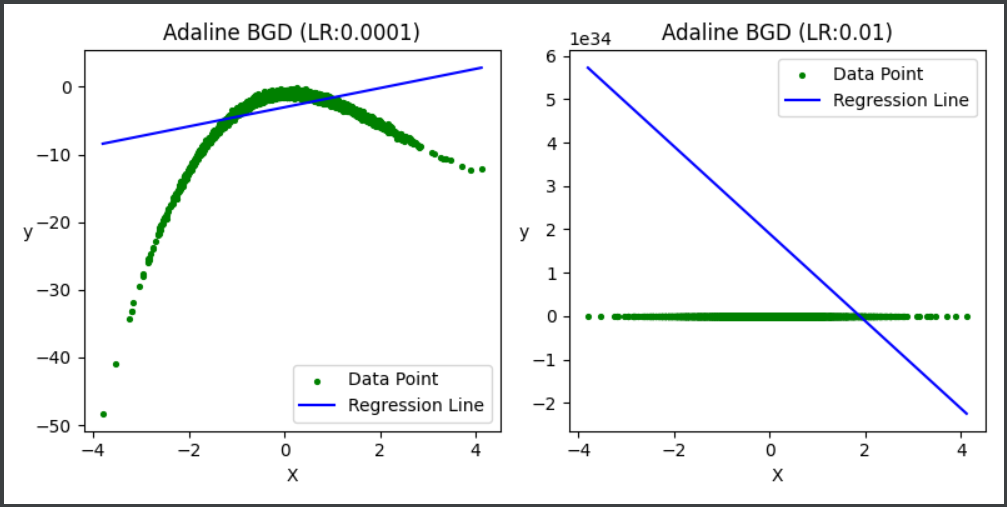


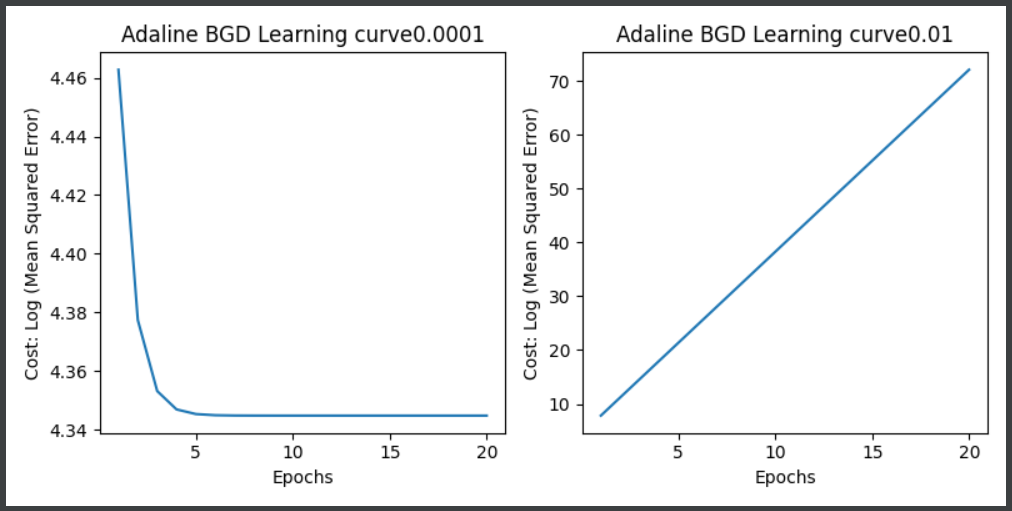
**Output:**

In the output, you can see the predicted weights, learning rate, intercept, Slope, and mean-squared-error values respectively:

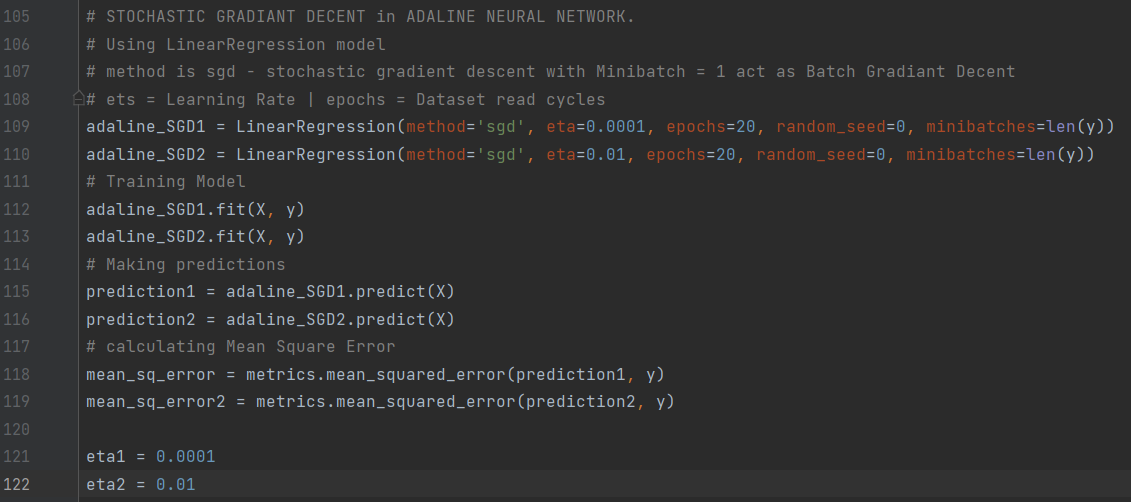


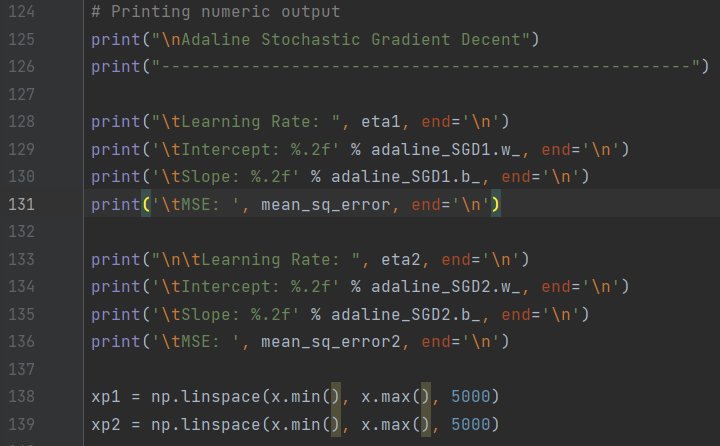
The plots helps us visualize our hyper-parameters in more details and easily.



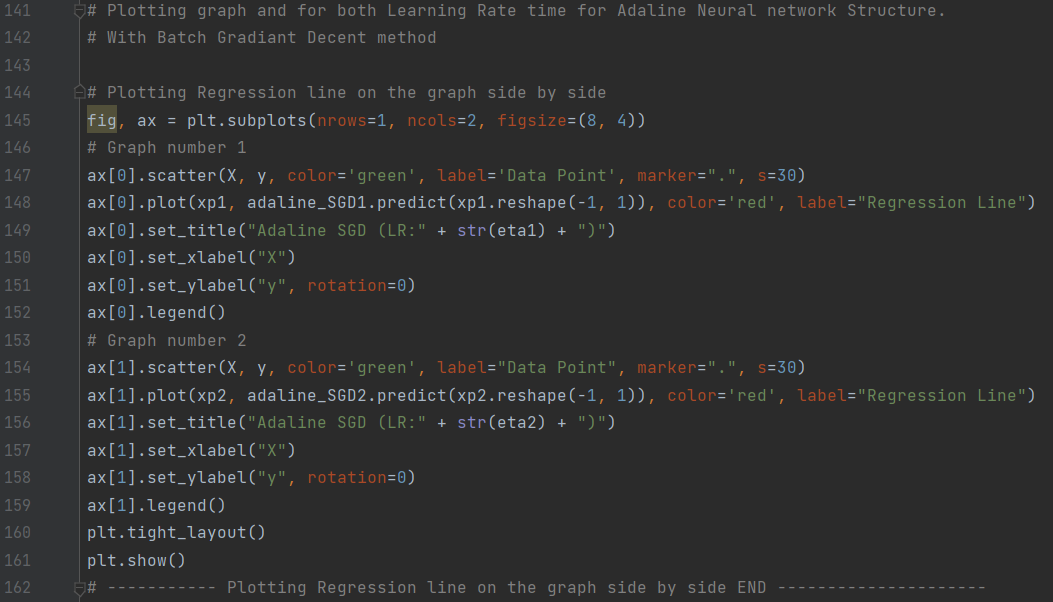


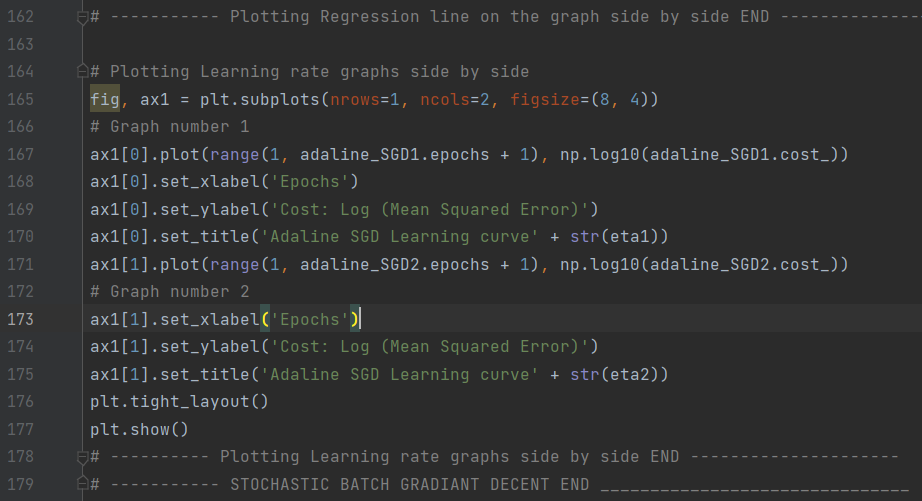
1. In this part, we are implementing Adaline Learning Algorithm using stochastic gradient descent [SGD] formula.





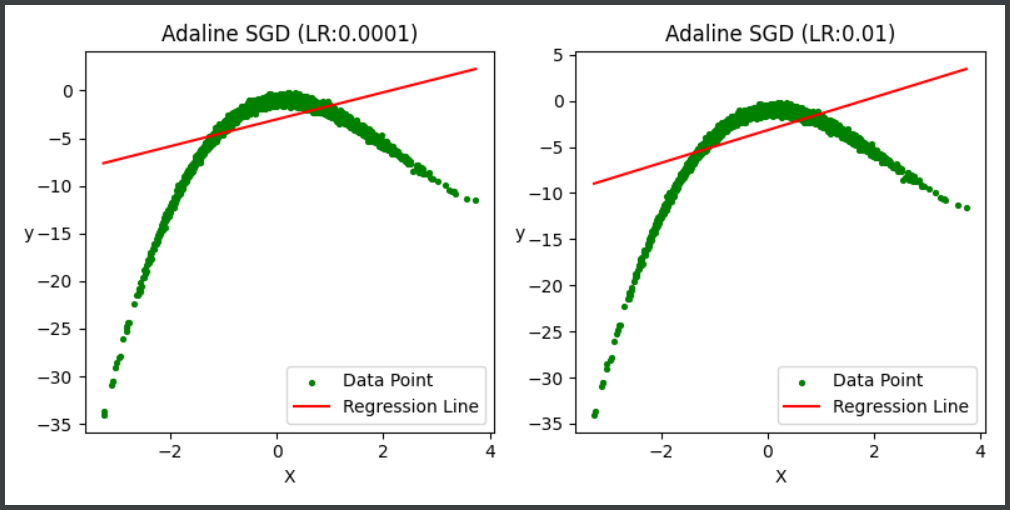
Now, we run the algorithm on our synthetic dataset.

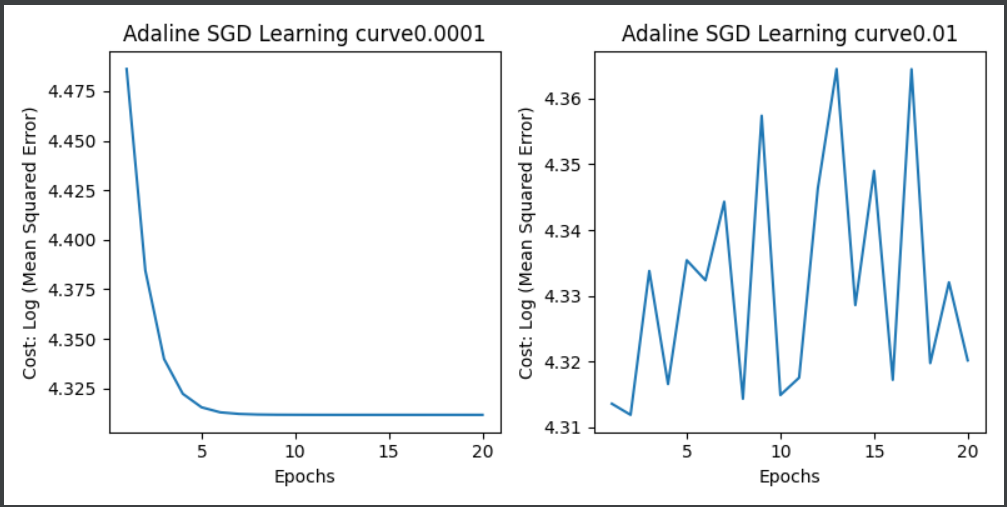




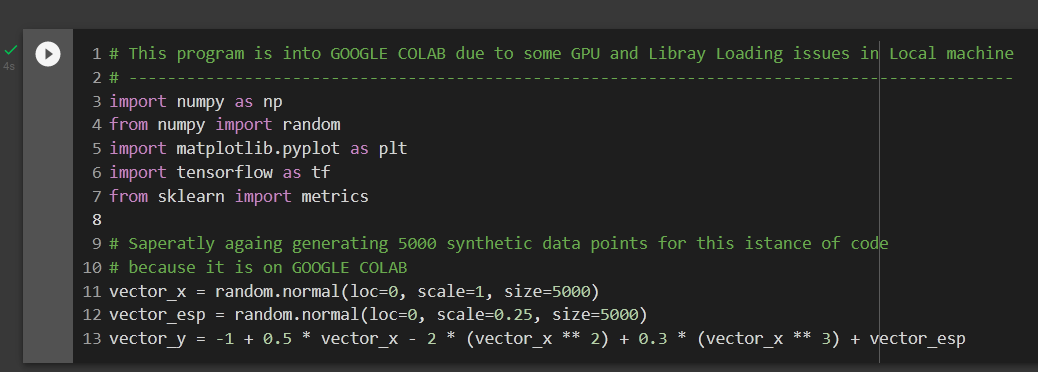
**Output:**

As you can see our predicted weights, learning rate and MSE values are shown below in a comparative plot.

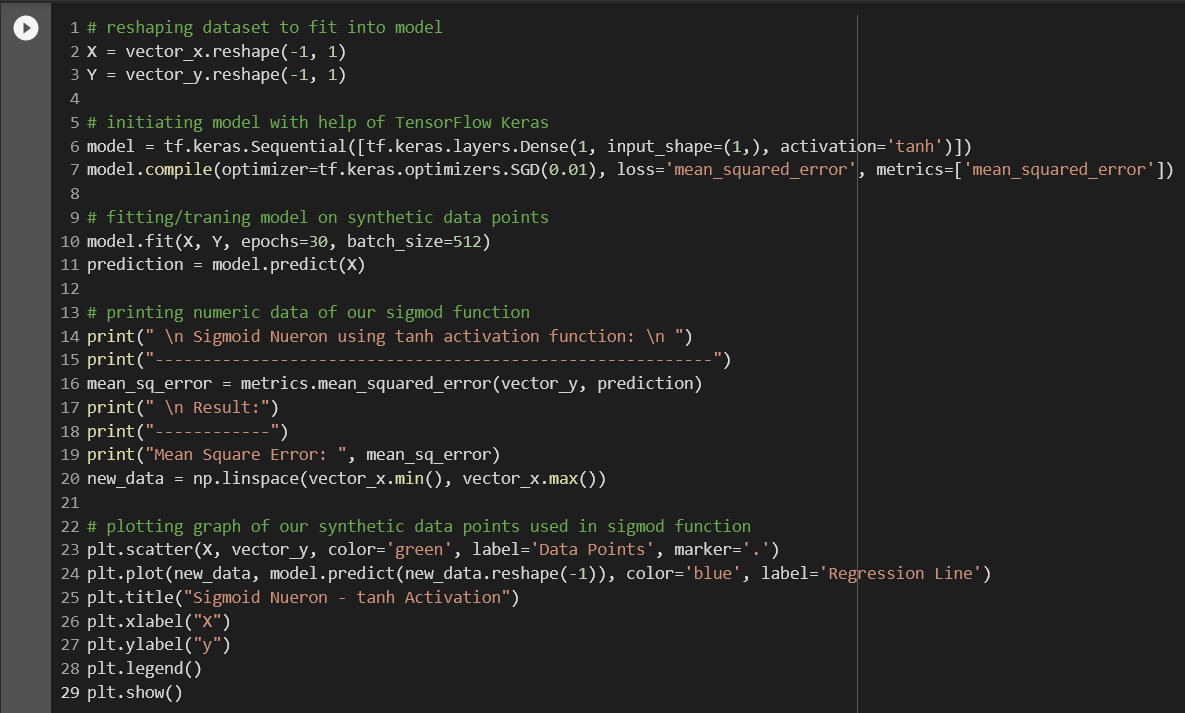




1. In this part we are going to Implement a Sigmoid Neural Learning Algorithm with a tanh activation function. Below is the code for that:



Now, running the algorithm on our synthetic dataset.

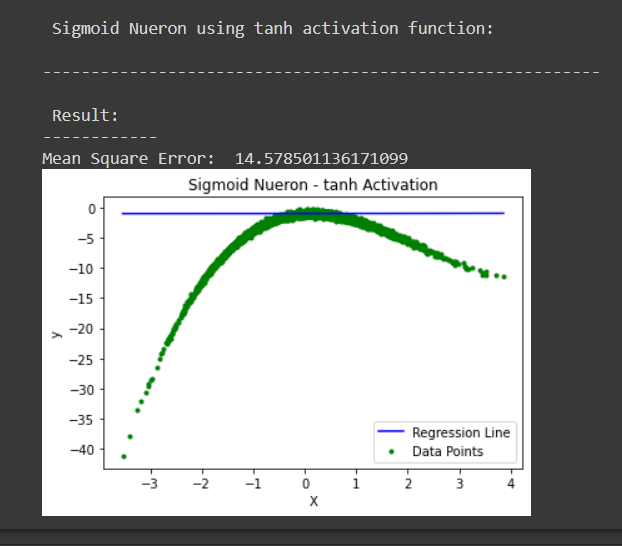


**Output:**

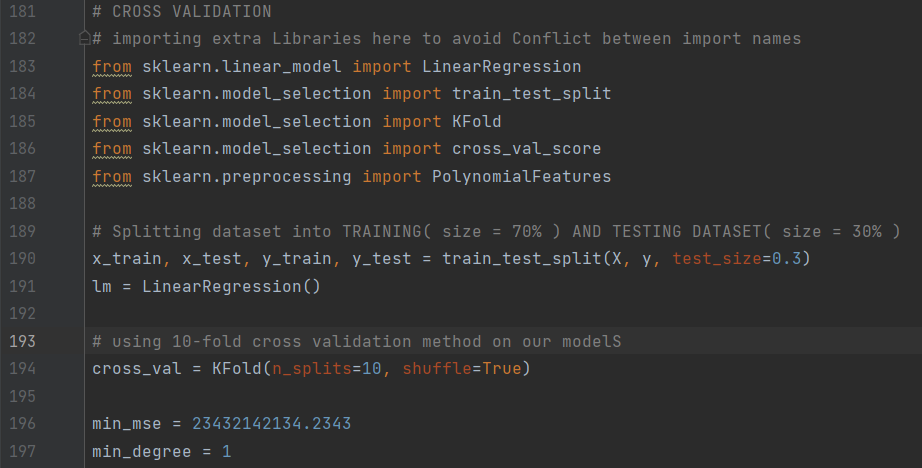
The predicted values of weight and MSE is shown in the plot below.

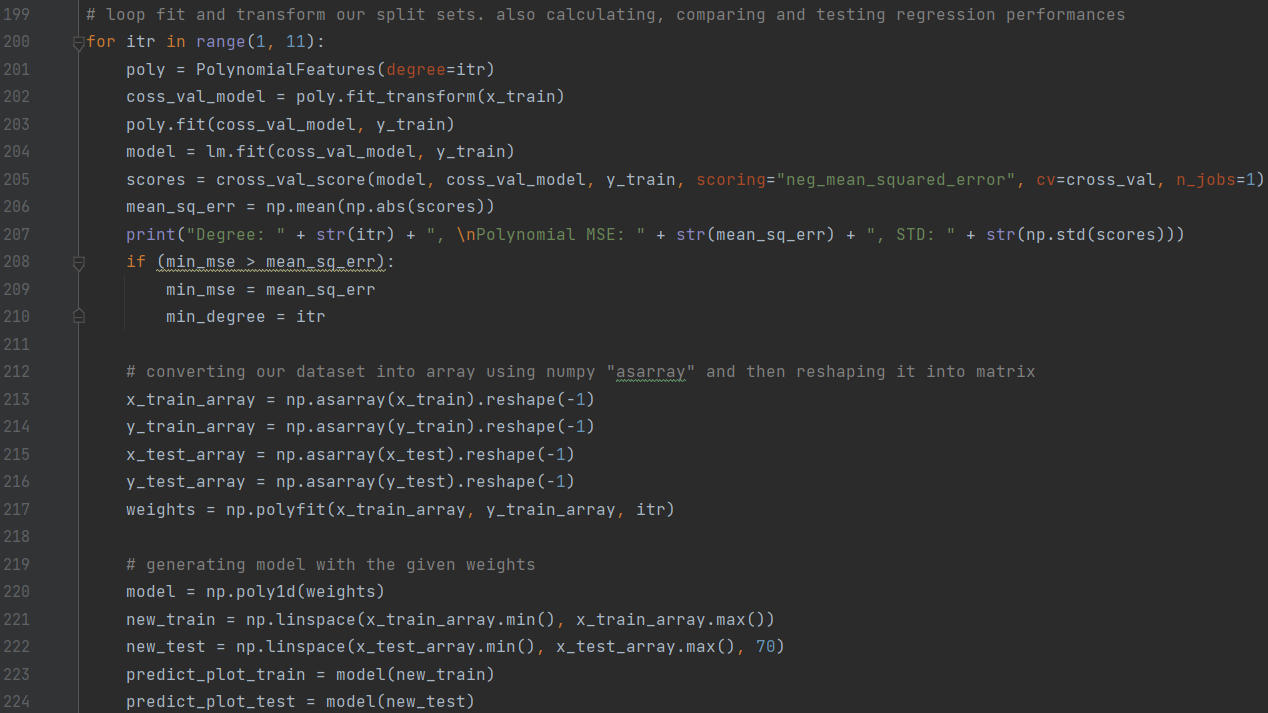
**Note:**

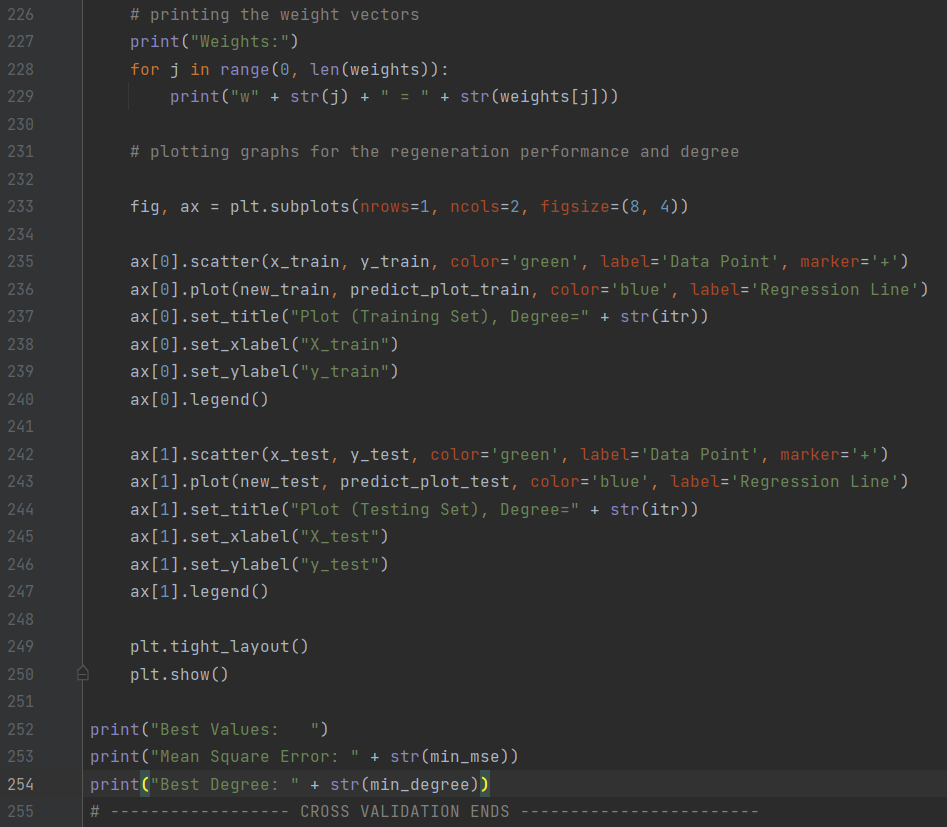
As seen, the sigmoid neuron does not give the correct output for our created dataset as the MSE value is very high.



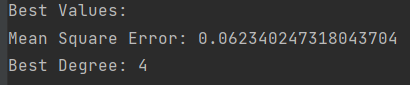
1. In this part, we are performing cross validation(10-fold-CV) of Batch gradient descent algorithm and the stochastic gradient descent algorithm.

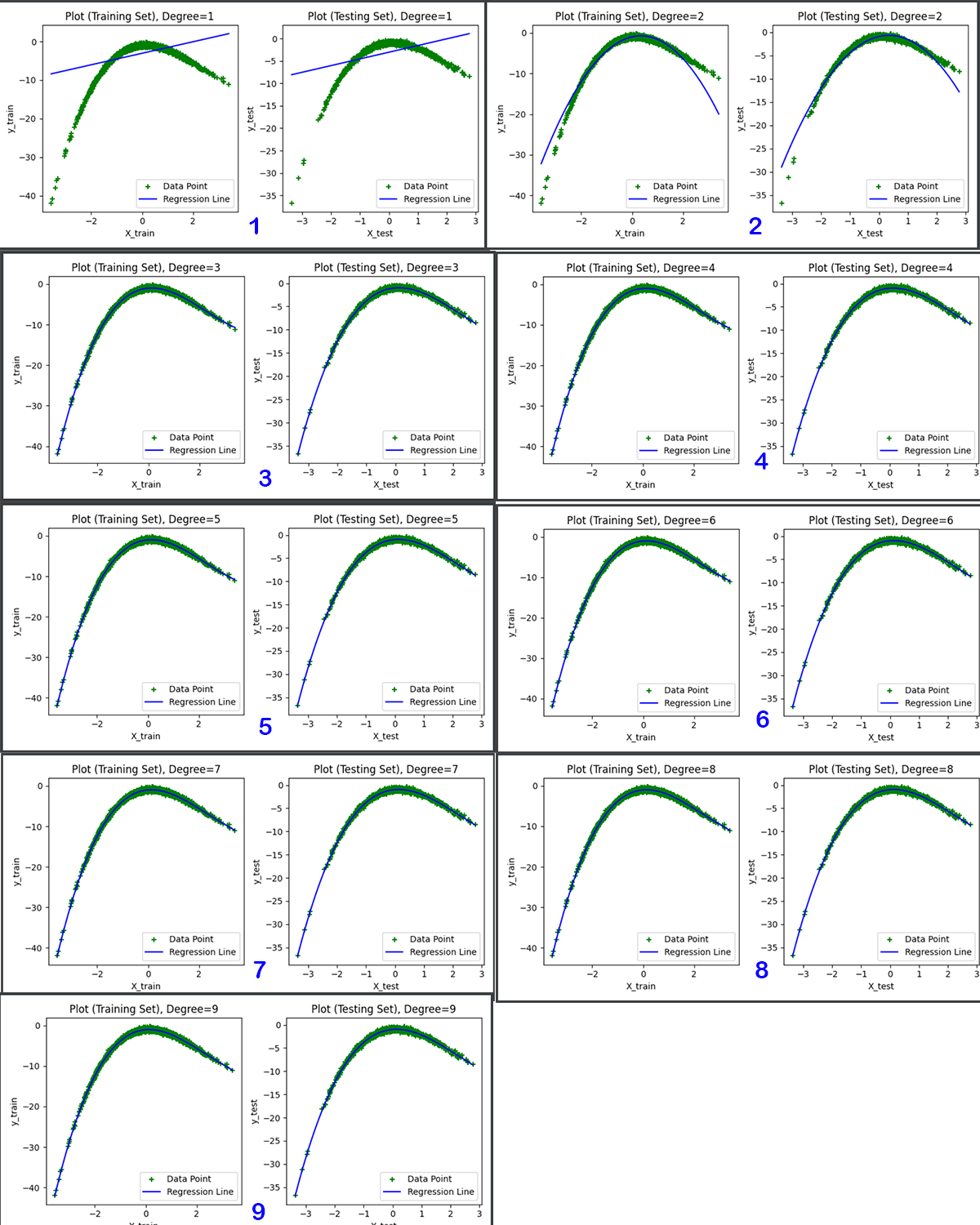






**Output:**





**Observation:**

During the experiment, we tried different values of learning rates and epochs to check their effects on the performance of the given algorithms. With many attempts of trail and errors, we concluded the optimal values for our experiment.

We implemented BGD it is found that learning rate is too small in this method that is 0.01.

Before implementation of cross validation on Adaline and sigmoid the mean square error is high. After implementation of 10-fold cross validation method the mean square error is deduced to 0.06234024731. So, it can be said that, the solution is more optimized after implementing cross validation.

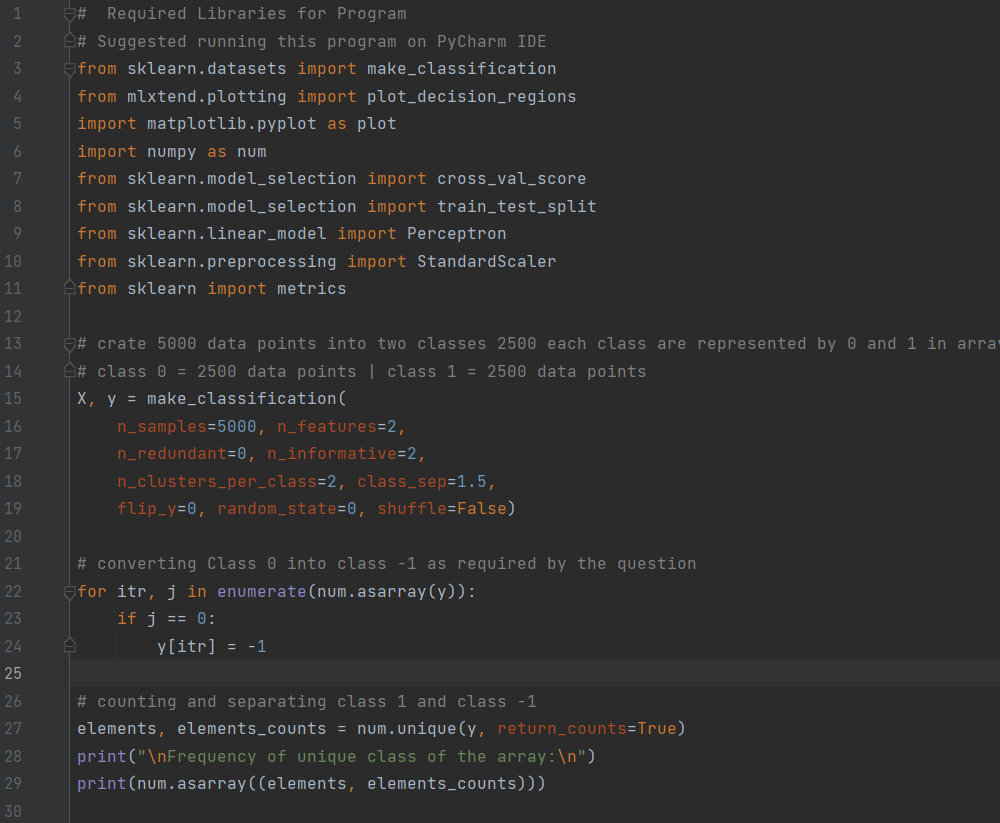
**Question: 2 In this question you are to create some simulated data sets and then use the Perceptron neuron to perform some classification.**

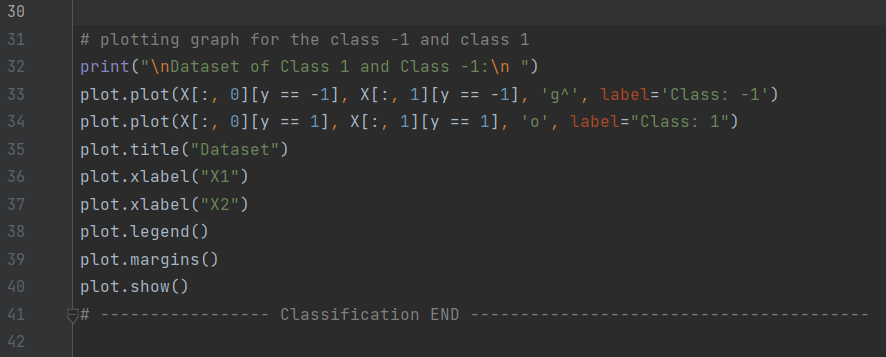
* **Randomly create 2500 data-points (x, y)’s of class -1 to lie one side of the function f above and 2500 data-points (x, y)’s of class +1 to lie on the other side of the function. Indeed, here, you are not required to create your data using the function f above; you can use any function you want, as long as it is a simple linearly separable function of your choice to be used to separate 5000 data points into two classes (I have mentioned the function above simply because you have it already).**

* **Implement the Perceptron learning algorithm and run it on your synthetic data set. Obtain the best Perceptron model via any cross-validation method of your choice. Use your creativity to tell me anything about your Perceptron: for example, how does the performance (speed and accuracy) vary when changing the learning rate, or when varying the size of the size of the training and test sets?**

**Solution**

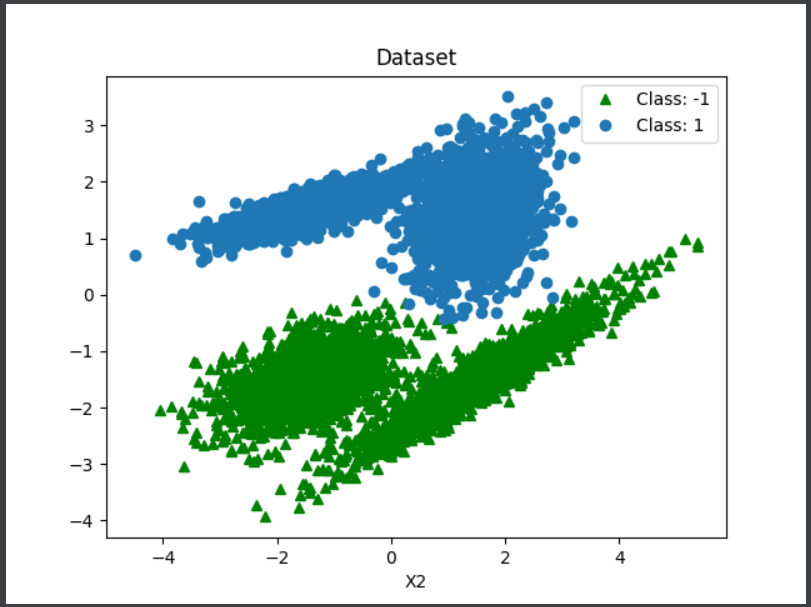
1. Import all the required libraries and randomly create 2500 data-points (x, y)’s of class -1 to lie one side of the function f above and 2500 data-points (x, y)’s of class +1 to lie on the other side of the function.



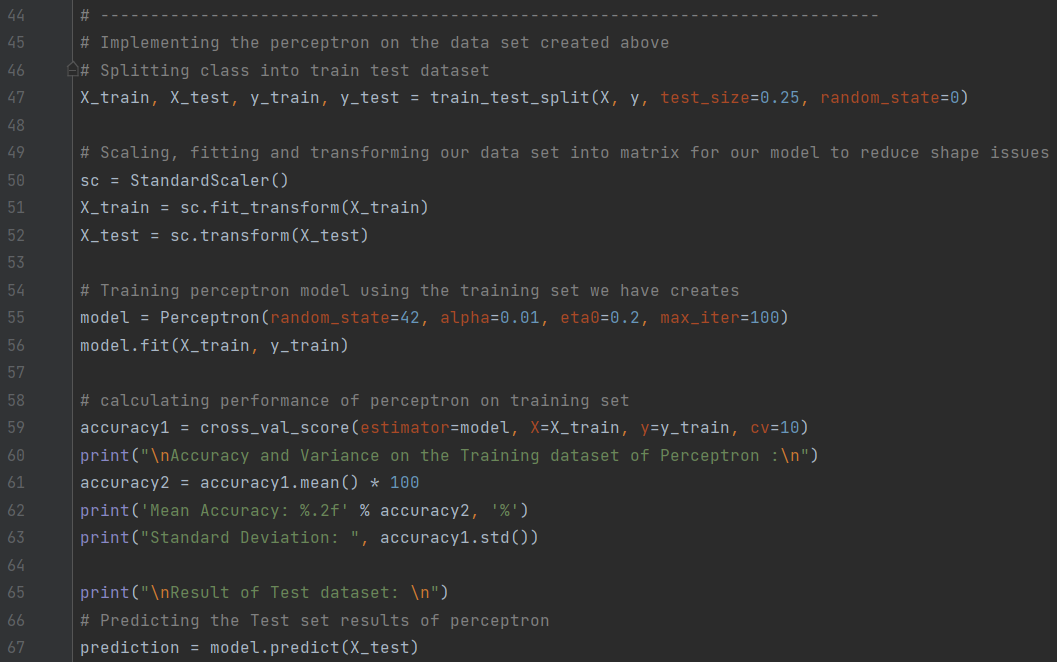


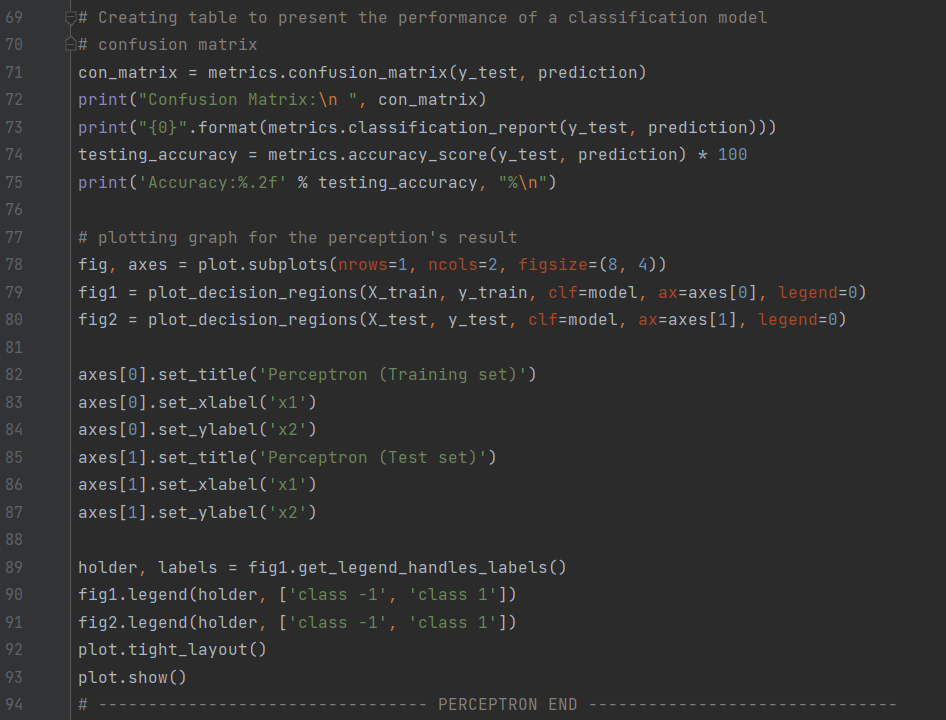
**Output:**

We can see a graph of separate data points of blue and green colour.



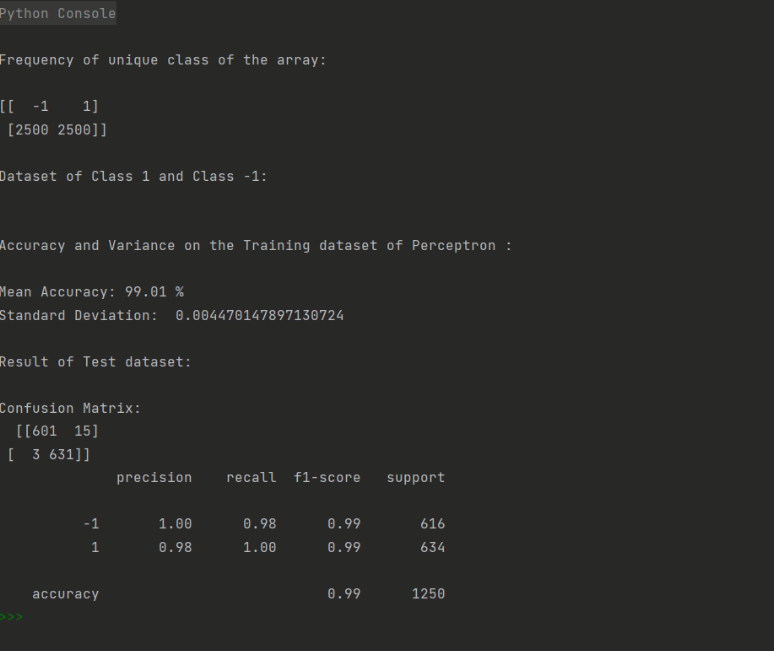
1. In this part, we are implementing Perceptron learning algorithm to obtain the best Perceptron model via cross-validation method

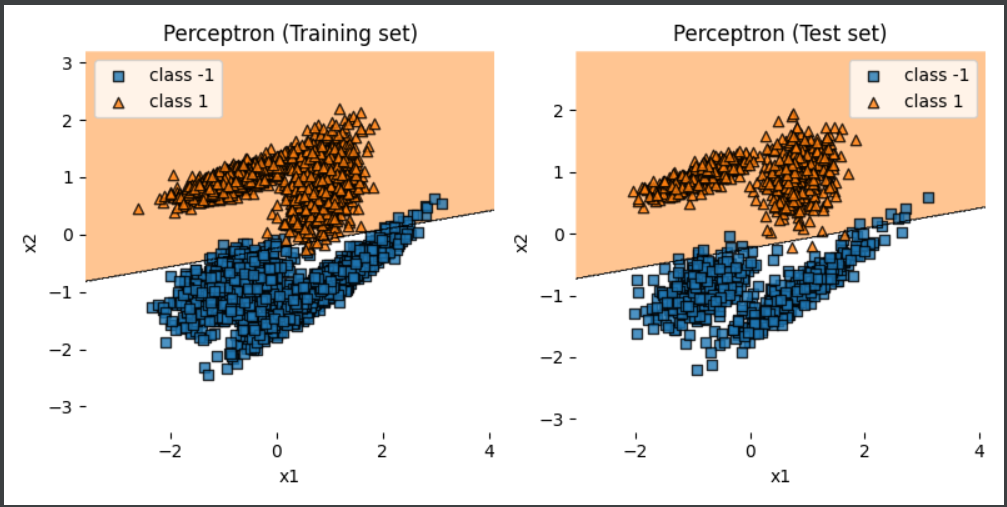




**Output:**

As seen below, the frequency of our unique class of the array is shown below. Also, we have achieved an accuracy of 99.01% with a standard deviation of 0.004 value. The results on the test datasets along with a confusion matrix is given below:





By using the perceptron learning algorithm datapoint has been separated in -1 and 1. After splitting the dataset in train and test data and implementing the CV method the data set become more accurate. As shown in output above the data point is plotted below and above the slope more accurately i.e. 99.01%.

Thank You