**Gradient Descent Algorithm**

**(With and without Step Function)**

Max iterations = 10,000

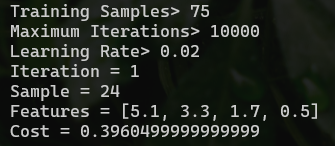
Training Sample = 75

Testing Samples = 75

Learning Rate = 0.02

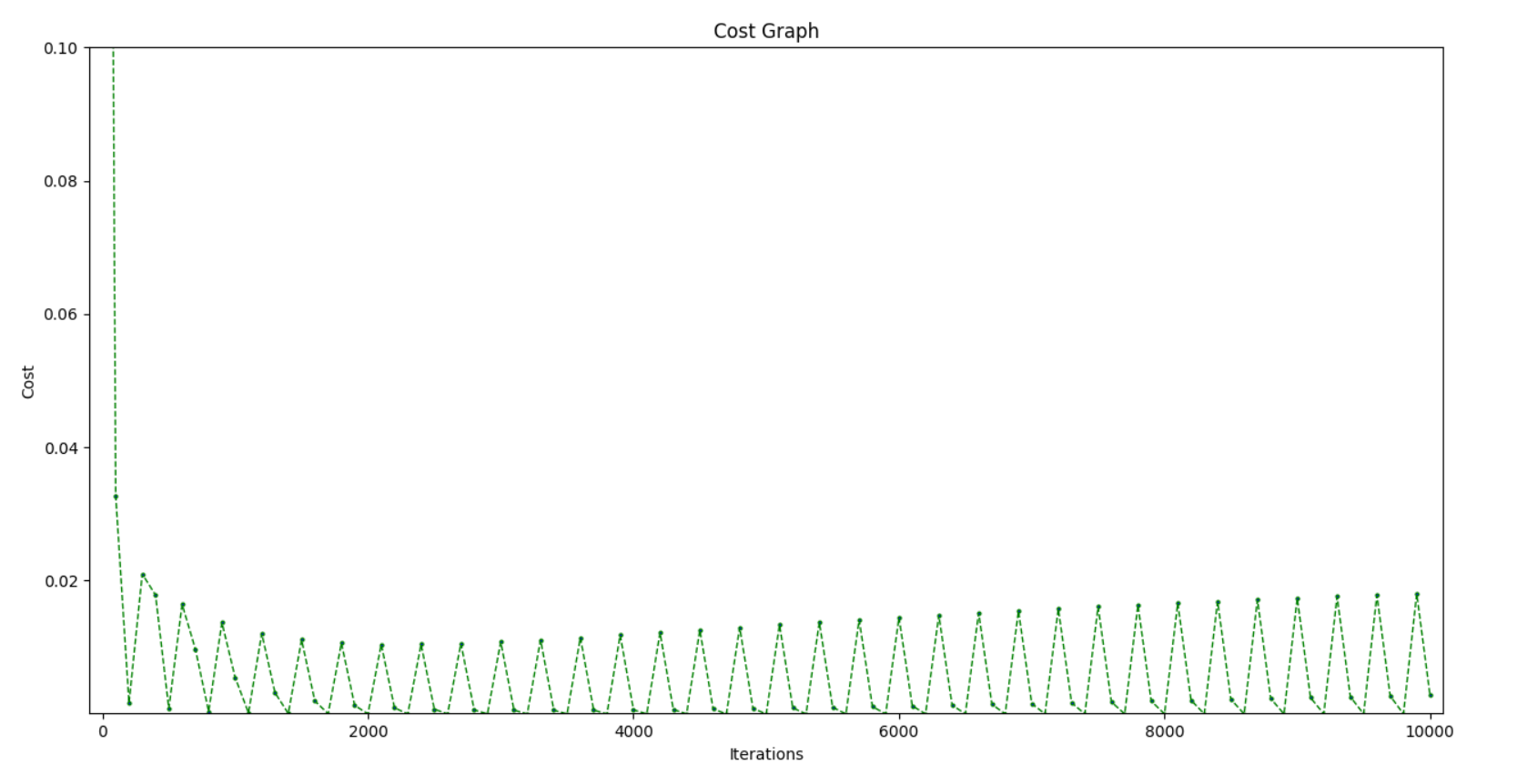
Initial Weights = [-0.2, -0.1, 0.1, 0.2, 0.3]

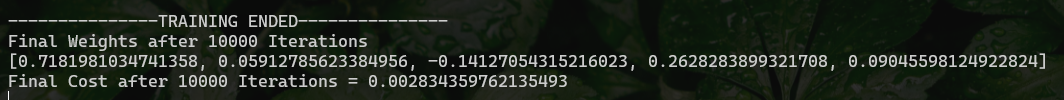
**Results:**



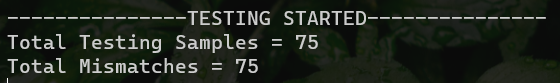
Cost after 1 Iteration = 0.396

Cost after Max Iterations = 0.0028





**Testing Phase without Step Function**

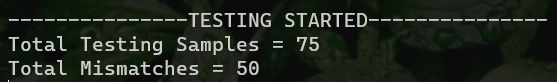


**Problem**

The problem which arises when we don’t use a step function is that we have assigned whole number values to y (actual class label). But when we calculate y\_hat (predicted class label), the model returns continuous values.

Now whenever we calculate the difference i.e. y – y\_hat, it never matches even if y was 1 and y\_hat was 0.9. This is why we have to use Step function to convert y\_hat’s value to some whole numbers.

**Testing Phase with Step Function**

****

The results did improve as the number of mismatches decreased. But seems like the model’s accuracy is not that good with 75 samples as learning samples.

Accuracy = 33.3 %

**Changing Percentage of Training and Testing Samples**

Max iterations = 10,000

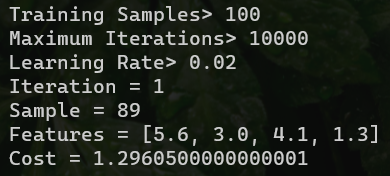
Training Sample = 100

Testing Samples = 50

Learning Rate = 0.02

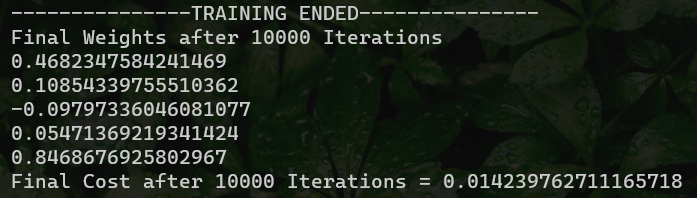
Initial Weights = [-0.3, 0.2, 0, -0.2, 0.3]

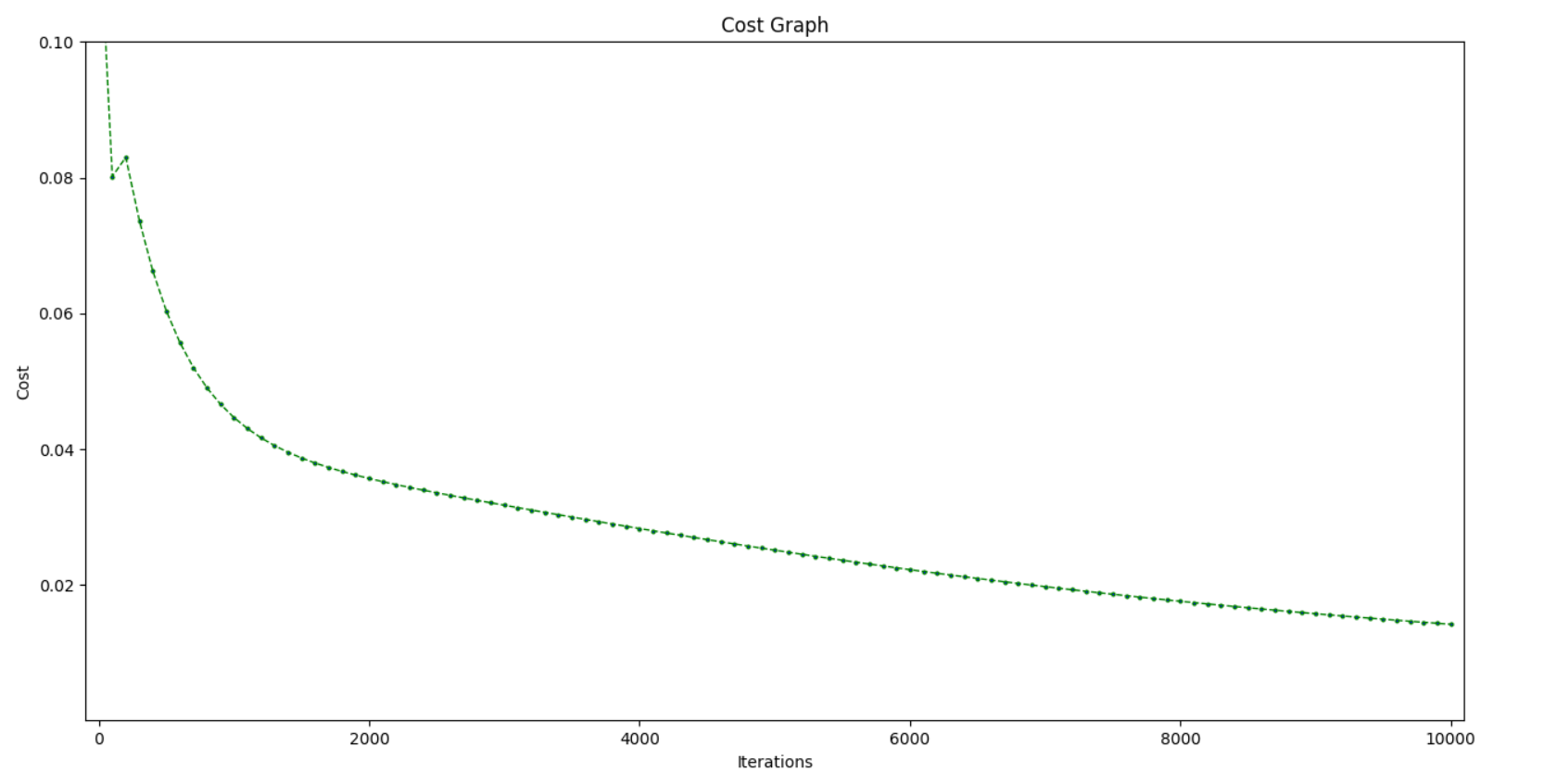
**Training Phase**

****

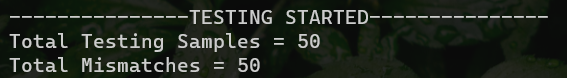
Cost after 1 Iteration = 1.296

Cost after Max Iterations = 0.0142

****



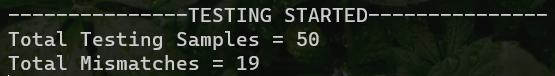
**Testing Phase (without Step Function)**



**Problem**

The problem is again the same as before. We have assigned whole number values to y, but the model returns continuous values as y\_hat.

**Testing Phase with Step Function**

****

The results did improve as the number of mismatches decreased.

Accuracy = 62 %