CS 487 - HW2

Cyrus Baker
February 22, 2020

Introduction

The purpose of this assignment is to implement three different machine learning algorithms:
- Perceptron
- Adaline
- Stochastic Gradient Descent

Each algorithm was implemented as a module which was imported into the HW2.py file. Included in the zip file are all of the python files, datasets, and a readme.txt file to explain how to run the programs. The classifiers were modified with lines of code in order to plot errors and cost values and accuracy scores over all of the iterations. The accuracy of each iteration is printed to the console. Because some methods had 50 iterations I decided not to include them in a table.

Description of Datasets

For my experiments I used two different datasets:
- iris.data
- wine.data

The iris dataset has 4 attributes, sepal width, sepal length, petal width, and petal length. There are 3 unique target values, Iris-setosa, Iris-versicolor, and Iris-virginica. This dataset contains 150 instances, 50 of each. In order to do binary classification on this dataset I set any row with a target of Iris-setosa to 1 and all other rows to -1.

The ionosphere dataset has 34 attributes and 1 target column. The dataset contains 351 instances. Similar to the iris dataset I set any target with a value of bad to 1 and good to -1.

Running Times

Running Times in seconds		
	iris.data	ionosphere.data
Perceptron	0.35015555100108	0.384238055994501
Adaline	0.189922852994641	0.147115293992101
SGD	0.195379573997343	0.143702872999711

Convergence

Convergence		
	iris.data	ionosphere.data
Perceptron	yes	no
Adaline	yes	yes
SGD	yes	no

See graphs below for the convergence of these algorithms on these datasets.

Feature Scaling

I used feature scaling for all models on all datasets in order to have values that range between zero and 1.

Learning Rates

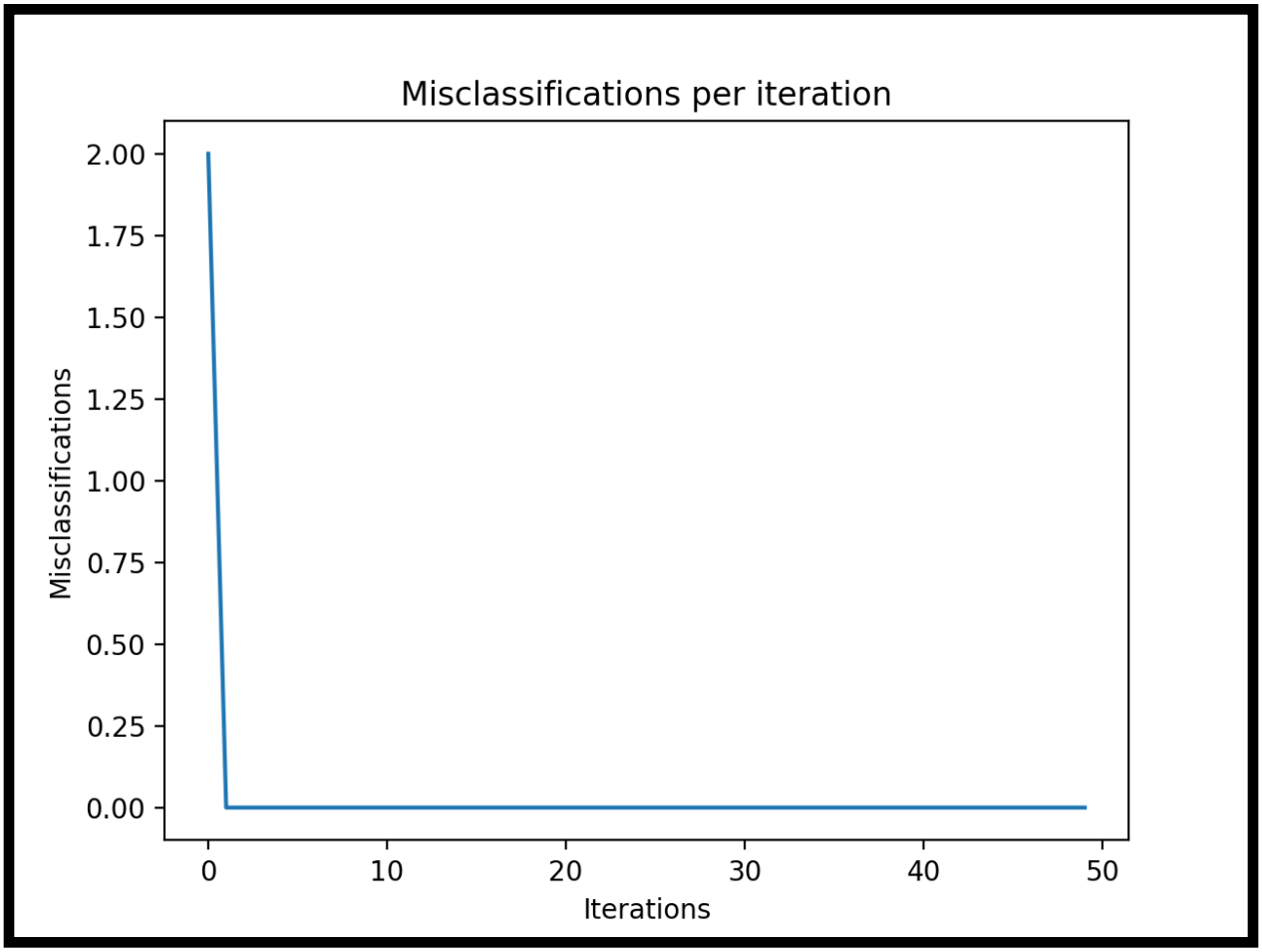
Learning Rates		
	iris.data	ionosphere.data
Perceptron	0.01	0.01
Adaline	0.001	0.0001
SGD	0.01	0.01

Number of Iterations

The perceptron and Adaline methods went through 50 iterations each while the SGD only went through 10.

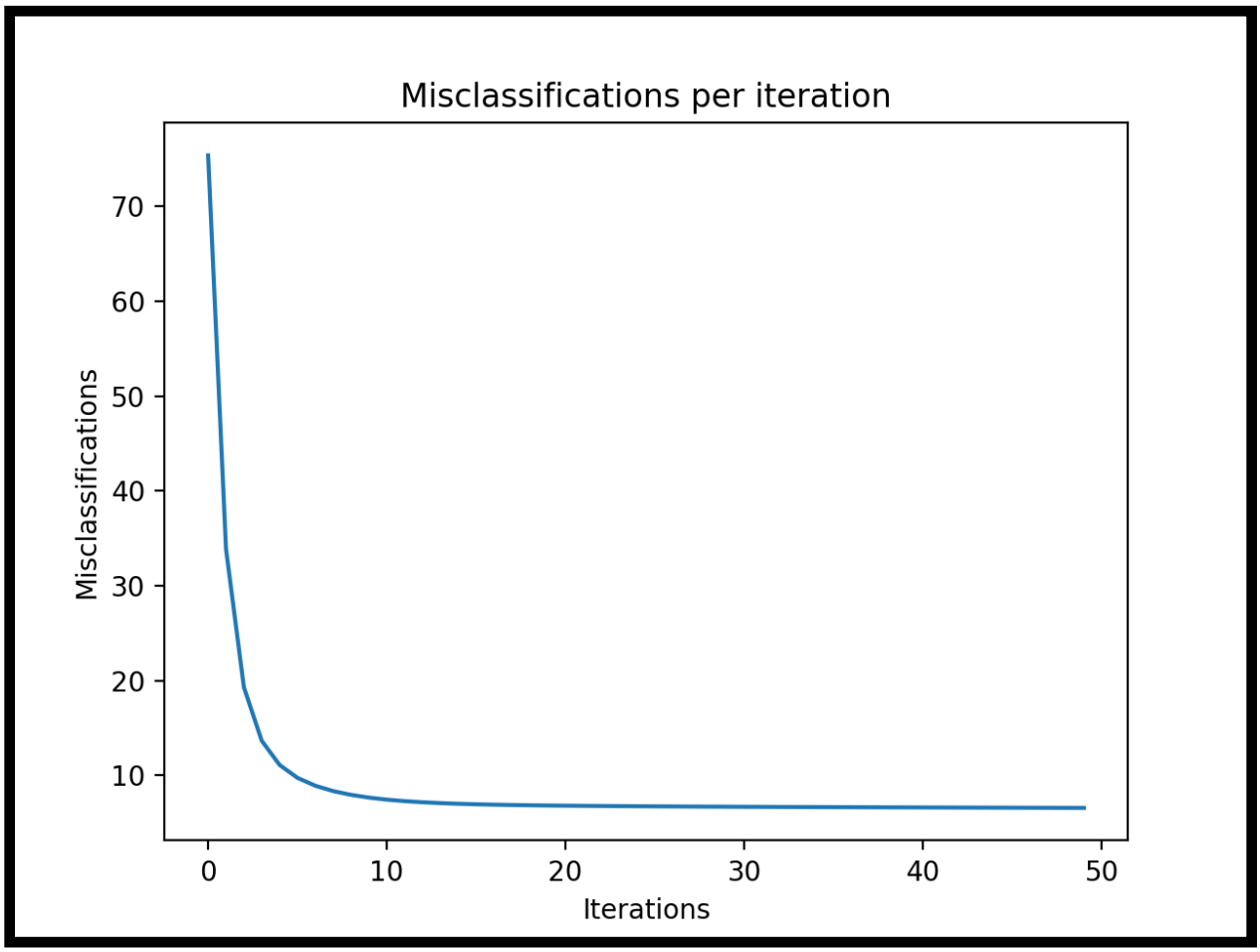
Plots

Iris Perceptron



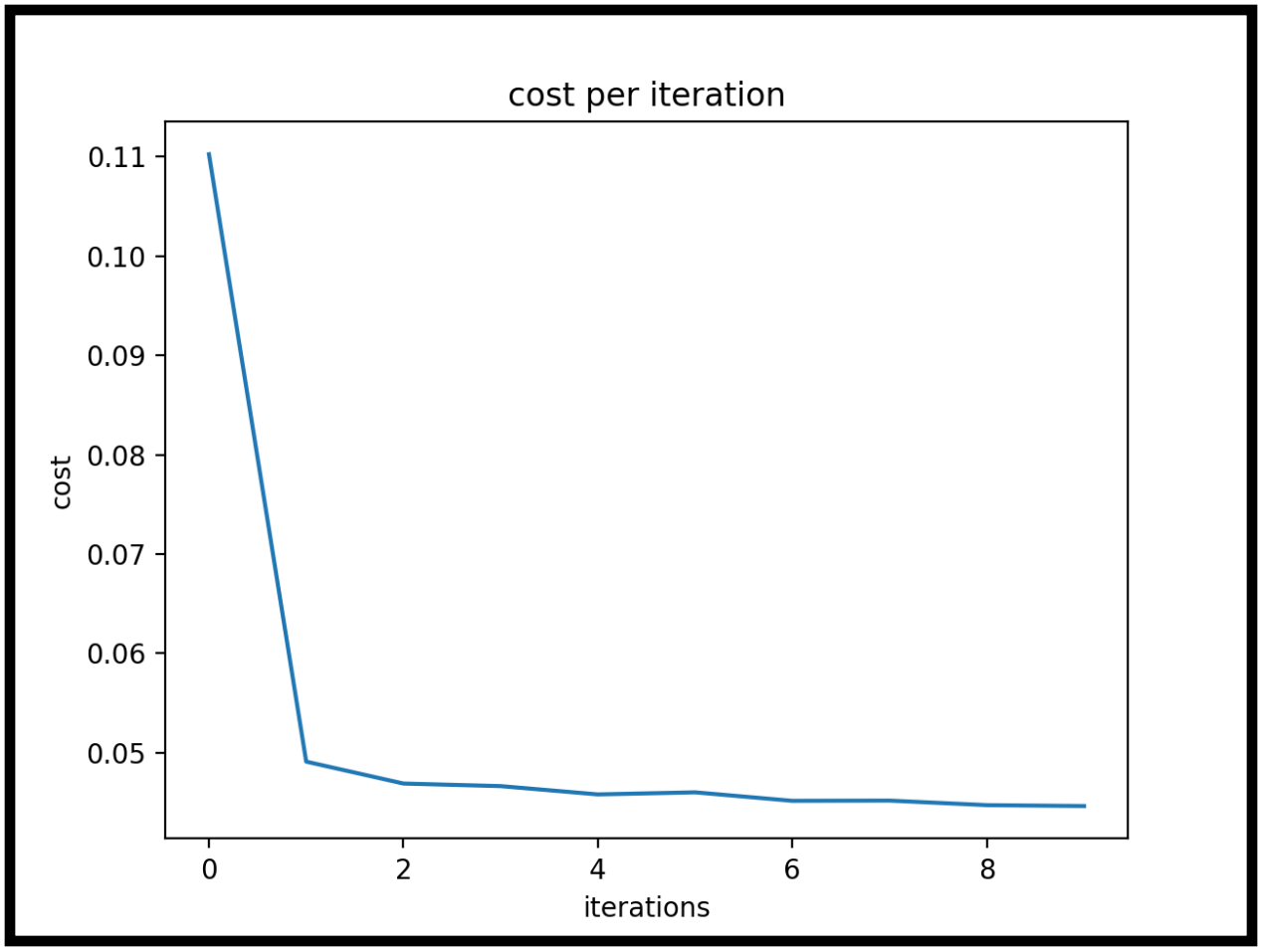
This method converges for this data set very quickly which means it is a good classifier.

Iris Adaline



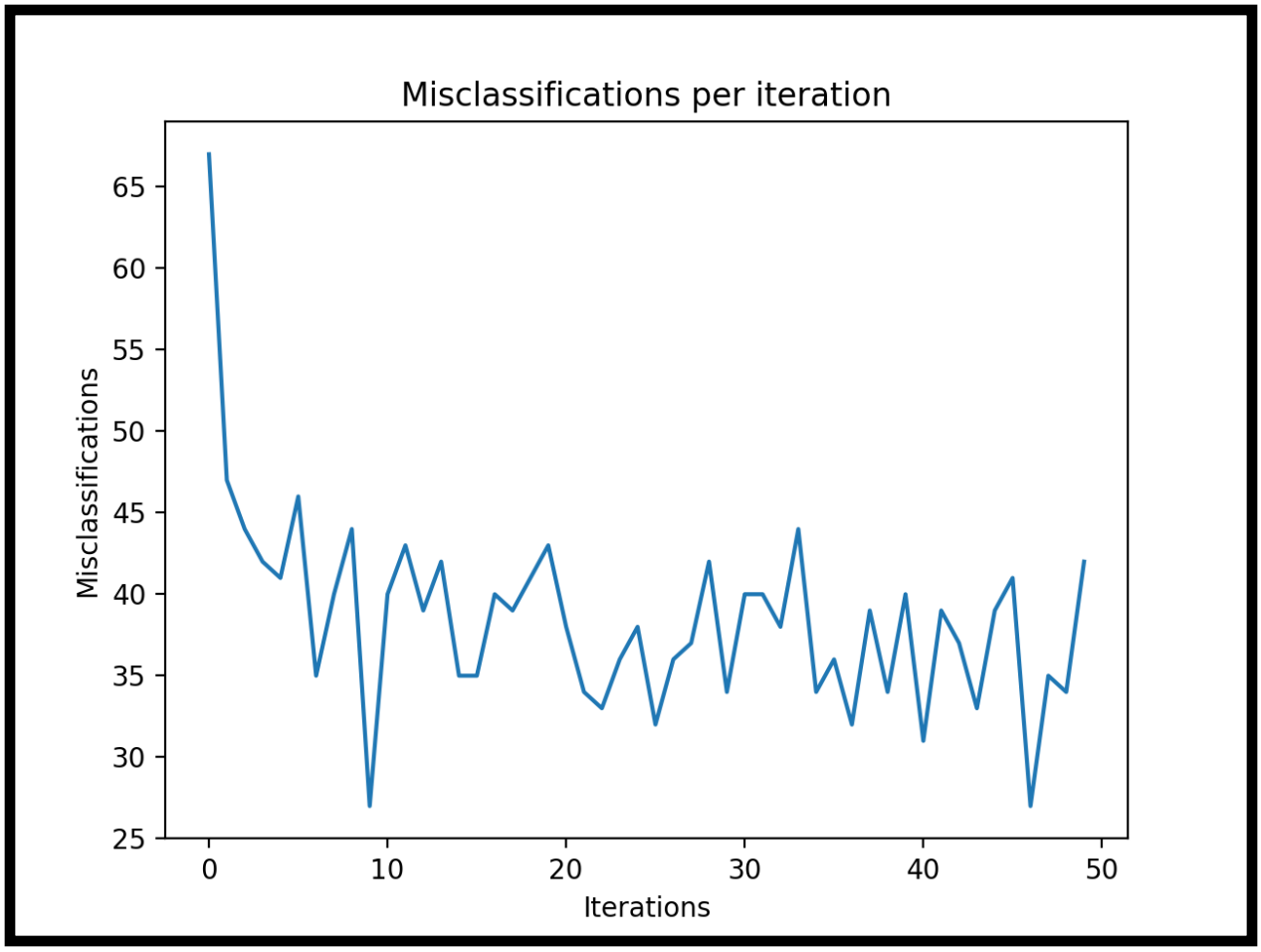
The Adaline algorithm also converges for the iris data set but it takes more iterations to do so.

Iris SGD



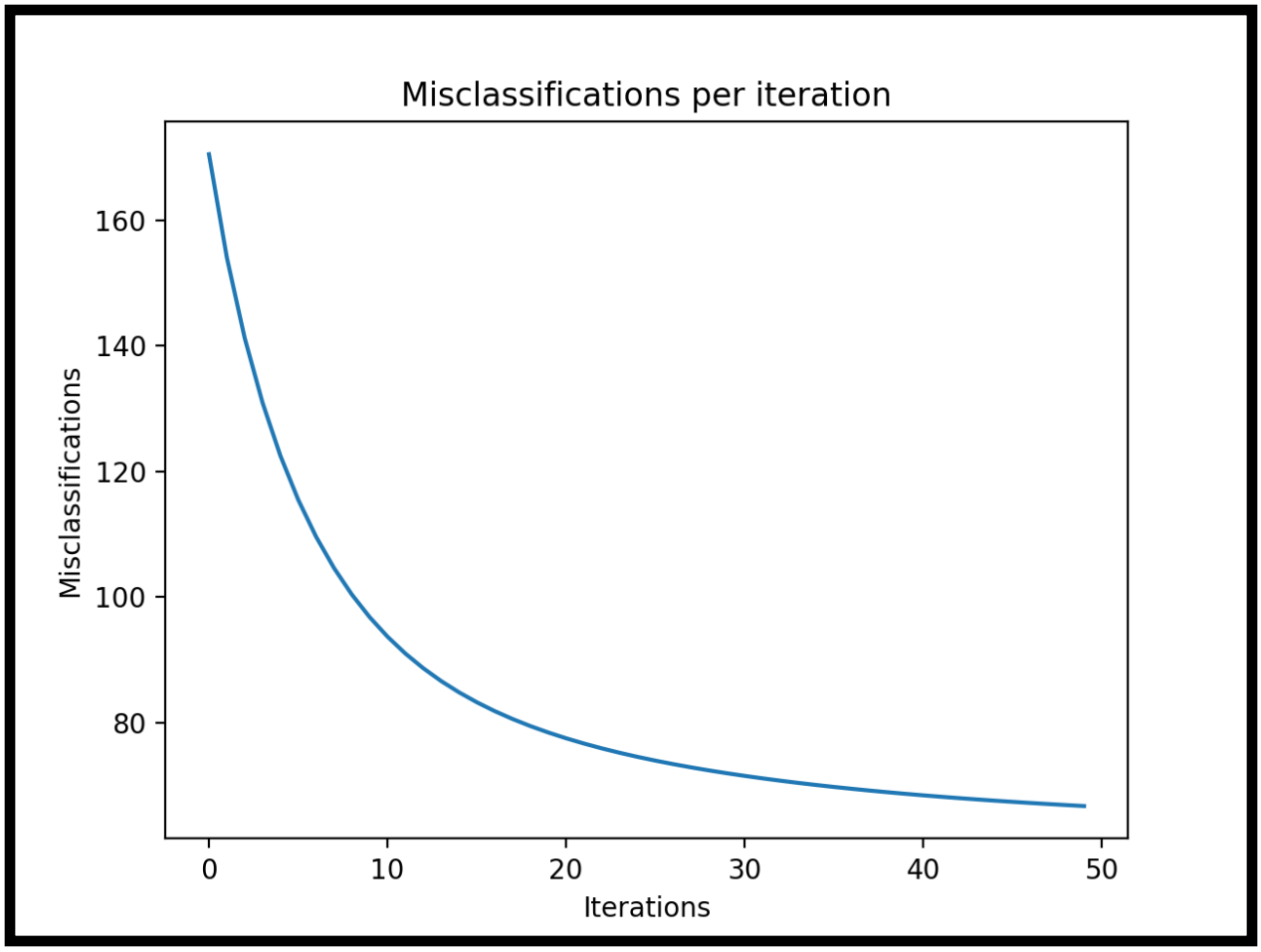
The SGD algorithm would also be a good classifier for this dataset because it converges after only a few iterations.

Ionosphere Perceptron



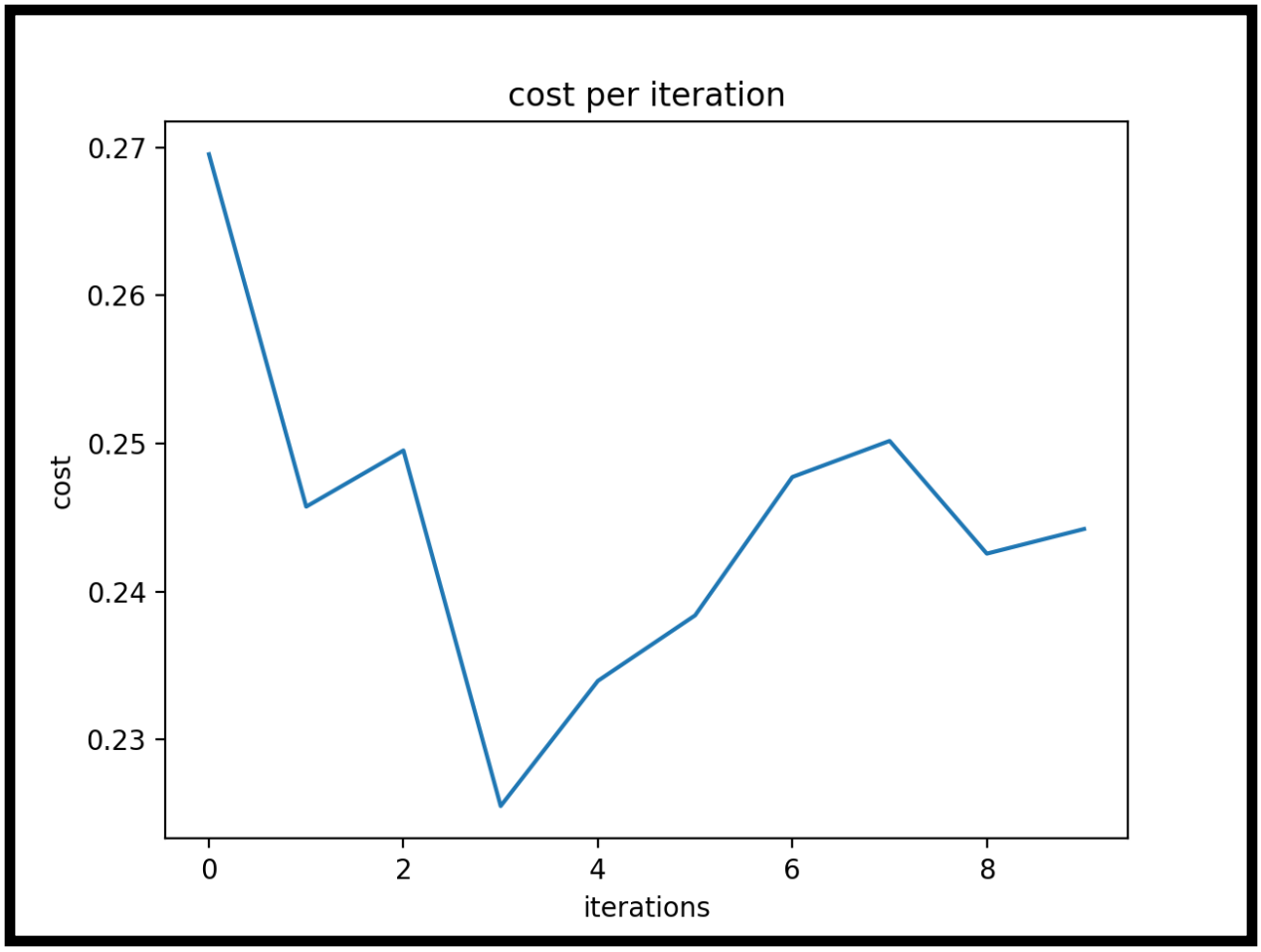
The perceptron model appears to have a pattern where it is making fewer classification errors as the number of iterations increase but after 50 iterations it is still making a significant number of misclassifications and it is not converging.

Ionosphere Adaline



For the ionosphere dataset this classifier is the best option but the learning rate had to be reduced from 0.001 to 0.0001. But the error rate is still very high.

Ionosphere SGD



The SGD is not a good fit for this dataset because it does not have a clear pattern of convergence.