Jack Gallagher
AI
4/23/2024

Report

Objective: MLPClassifier used to identify whether a meteor is hazardous or non-hazardous

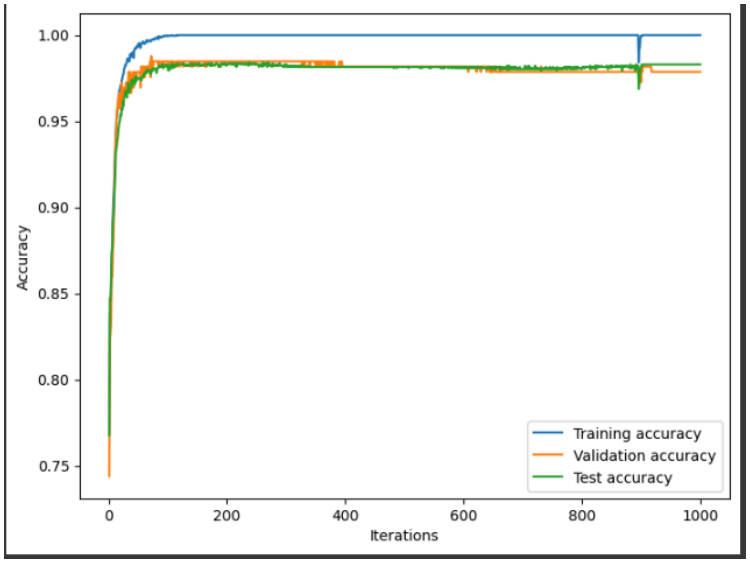
Setting up data for the experiment:

When setting up the data set it is important to note that I was able to grab the data through google drive by utilizing drive.mount and then reading to the copied files location in my drive. From that I then identified the x columns that would not hold no relevance to what we are trying to measure (dates, names, etc.). After making sure the proper columns were assigned, we then use the standard scale to make sure that the data used is on a balanced scale.

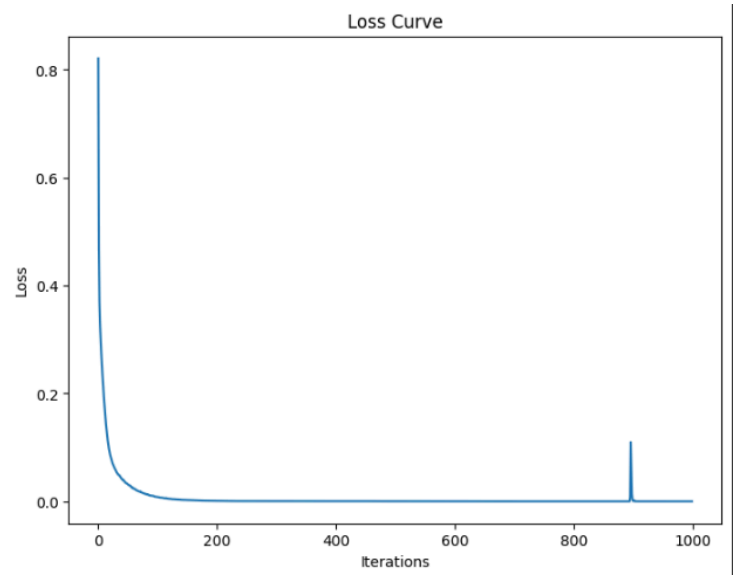
Test 1 (Base test):

- Hidden Layers- 2
- Neurons in each Layer- 50:50
- Iterations- 1000

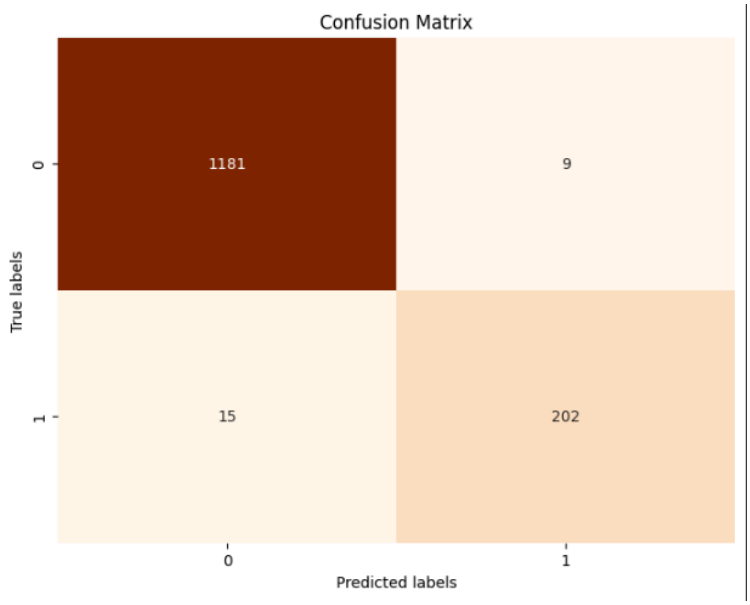
Training Validation and Testing Accuracy:



Loss Curve:



Confusion Matrix:



Overall Test Accuracy: 0.9829424307036247 or ~98.29%

Analysis:

Firstly looking at the Training Validation and Testing Accuracy Plot graph, we are able to determine a long spread of efficiency after around 100 iterations but smooths out the further along it goes for validation and test accuracy. A notable point in the graph occurs at around 900 iterations where a giant spike happens in both loss and accuracy graphs showcasing an outlier outcome. Focusing on the confusion matrix, we are able to determine that 1383 total cases

were predicted correctly while only 24 total were mislabeled. At an accuracy level of 98.29%, high accuracy can be determined but let's test out what changing other variables can do to the test accuracy.

Test 2 (Extra Layer):

- Hidden Layers- 3
- Neurons in each Layer- 50:50:50
- Iterations- 1000

Overall Test Accuracy: 0.9857853589196873 or around 98.58%

Test 3 (Higher Neurons):

- Hidden Layers- 2
- Neurons in each Layer- 80:80
- Iterations- 1000

Overall Test Accuracy: 0.9793887704335466 or around 97.94%

Test 4 (Lower Neurons):

- Hidden Layers- 2
- Neurons in each Layer- 10:10
- Iterations- 1000

Overall Test Accuracy: 0.9836531627576404 or around 98.37%

Analysis of different hyperparameters:

Based on the tests that I ran less using less neurons actually lifted the test accuracy and adding an extra hidden layer boosted test accuracy. Testing higher neurons actually led to a lower accuracy when compared to the base case. With this logic less neurons and extra layers will aid when finding accuracy. Notably when running the code a few times the accuracy percentages do fluctuate and the difference between each test is not anything drastic as it tends to be the difference of 1 to 5 astros being deemed accurate or not.

Additional Test of Changing Iterations:

Test 5 (Adding Iterations):

- Hidden Layers- 2
- Neurons in each Layer- 50:50
- Iterations- 5000

Overall Test Accuracy:0.9800995024875622 or 98.01%

Test 6 (Lowing Iterations):

- Hidden Layers- 2
- Neurons in each Layer- 50:50
- Iterations- 100

Overall Test Accuracy:0.9793887704335466 or 97.94%
Overall Training Accuracy:0.9996612466124661 or 99.97%

Analysis of different iterations:

The raising of iterations to 5000 actually had a lower test accuracy when compared to the base test. The 5000 iteration program also took an extremely long time to run emphasizing that too many iterations has no benefit and should be avoided. The low iteration test is the only test we ran that did not hold a perfect training accuracy but notably was extremely close in test accuracy to the 5000 iterations. The 100 iteration also ran extremely quickly even in comparison to the base test and others performed earlier. Notably it might be beneficial to run the tests at a lower iteration such as 150/200 as you won't lose accuracy but will be able to perform at a faster rate.

Table Summary:

Test #	Hidden Layers	Neurons	Iterations	Test Accuracy	Training Accuracy
Test 1	2	50:50	1000	98.29%	100%
Test 2	3	50:50:50	1000	98.58%	100%
Test 3	2	80:80	1000	97.94%	100%
Test 4	2	10:10	1000	98.37%	100%
Test 5	2	50:50	5000	98.01%	100%
Test 6	2	50:50	100	97.94%	99.97%

Notable Reference:
https://github.com/bforoura/AI/blob/main/Module6/Iris_MLP_Classifier.ipynb