Sarah Groark
Artificial Intelligence (CSC 362)
Assignment 7 Report

Assignment Details: Given the NASA asteroid data set, design and train an MLPClassifier model that can accurately predict an asteroid as hazardous or non-hazardous. The data set (CSV) consists of 40 features of information about 4687 asteroids covering shape/geometry of the asteroid, path, speed, its location in relation to the planet it is revolving around, and more.

Model Parameter Details:

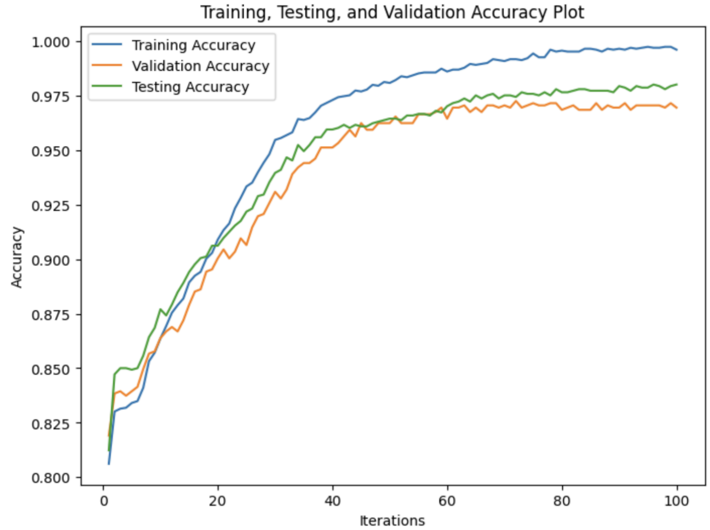
- Hidden Layers – 2
- Number of Neurons in each Hidden Layer – 15, 20
- Max Iterations – 100

Results:

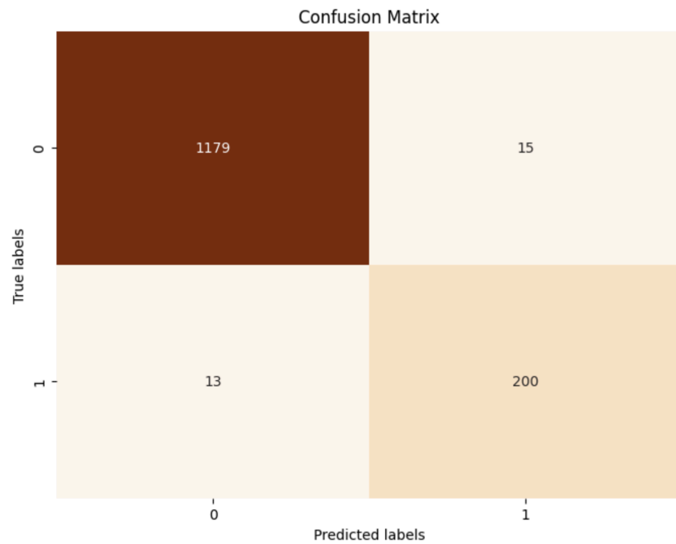
This MLPClassifier model was found to be a decent balance between underfitting and overfitting. The model was trained to classify asteroids, using their numerical data features, as hazardous or non-hazardous. The final accuracies following the training of the model are as follows:

Validation set	96.9%
Testing set	98.0%
Training set	99.6%

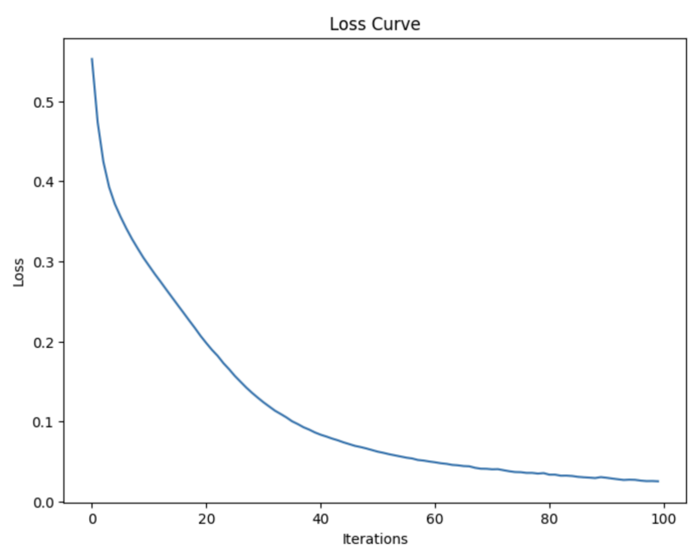
At the start of the iterations, the model fit resulted in accuracies falling between 80% and 82% for each of the three sets (testing, validation, and training). These values begin to converge around the 40th iteration, with values converging after passing approximately 92.5% accuracy.



The confusion matrix for this model shows that out of 4,687 asteroid examples, the model correctly predicted 1,179 of them to be non-hazardous, while it also correctly predicted 200 asteroids to be hazardous. Additionally, there were 15 asteroids that were predicted to be hazardous, but in actuality are non-hazardous. Similarly, there were 13 asteroids that were predicted to be non-hazardous, but were actually hazardous.



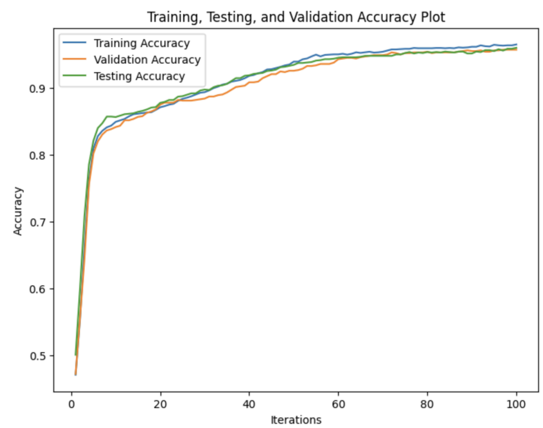
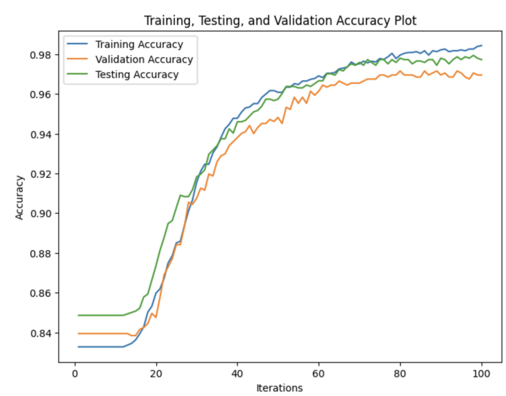
The loss curve plot is helpful in understanding how this model performs over the course of the iterations during which it is trained and tested. The loss function represents the difference between the predicted output and the true/actual output. Informally, the loss curve shows how far off the model's predictions are from the actual outputs over time. In this case, the model here proves to improve performance over the course of the experiment, as the loss decreases as the model moves through the iterations.

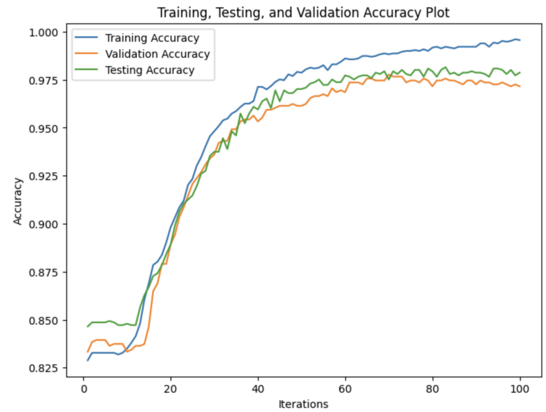


<u>Experiment</u>: observation of how changing hyperparameters affects the accuracy of the model

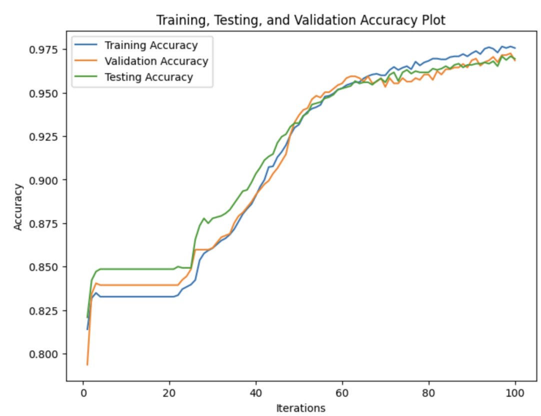
Hyperparameters to be tested –

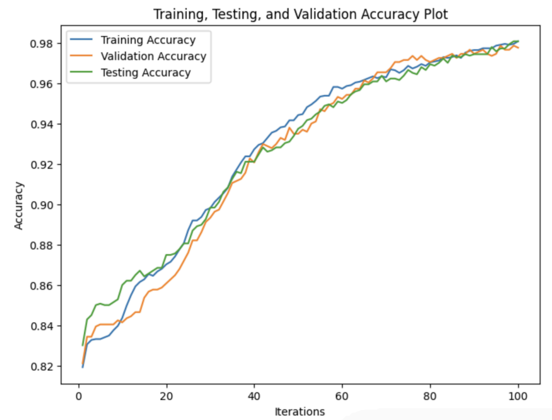
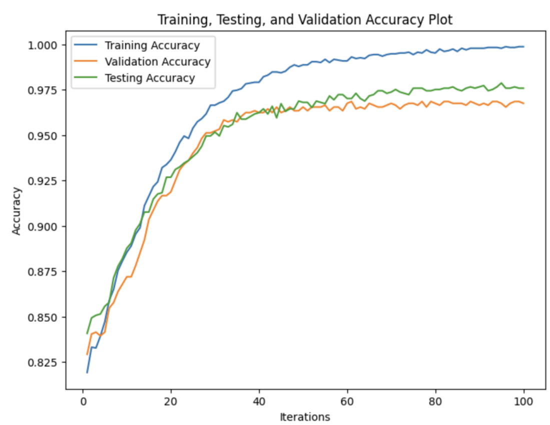
- Number of hidden layers (10 neurons per layer)

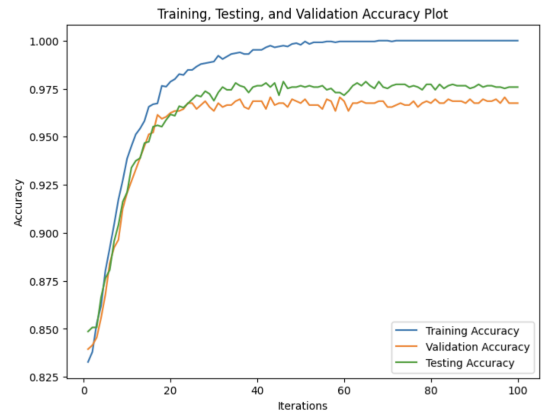
Hidden Layers	Neurons per Layer	Iterations	Accuracies (training, validation, testing)
1	10	100	96.8%, 96.7%, 96.0% 
2	10	100	98.3%, 96.9%, 97.5% 

3	10	100	99.5%, 97.1%, 97.9% 

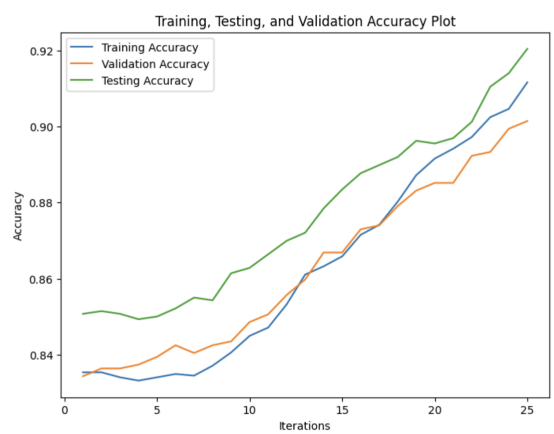
- Number of neurons per hidden layer

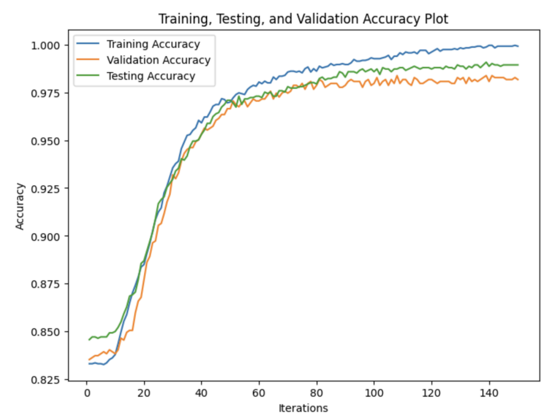
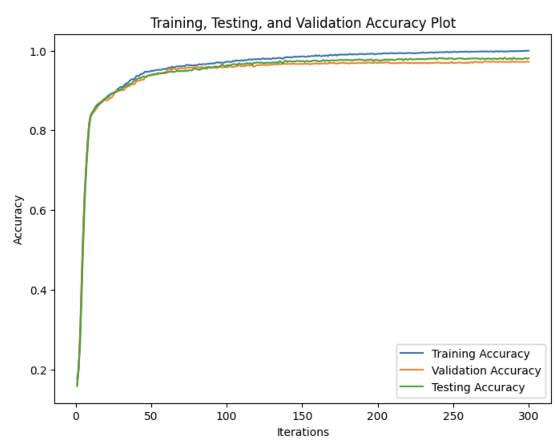
Hidden Layers	Neurons per Layer	Iterations	Accuracies (training, validation, testing)
2	3, 5	100	97.5%, 96.8%, 96.9% 

2	5, 10	100	98.0%, 97.7%, 98.0% 
2	20, 35	100	99.8%, 96.7%, 97.5% 

2	50, 75	100	100%, 96.7%, 97.5% 

- Number of iterations

Hidden Layers	Neurons per Layer	Iterations	Accuracies (training, validation, testing)
2	10, 10	25	91.1%, 90.1%, 92.0% 

2	10, 10	50	95.5%, 94.8%, 94.5% 
2	10, 10	150	99.9%, 98.1%, 98.9% 
2	10, 10	250	99.99%, 97.1%, 98.0% 

Observations –

The overall complexity of the data set determines how the hyperparameters should be adjusted to meet the desired requirements of the prediction model. In some cases, accuracy values of 100% indicate overfitting of the model, in which it is simply able to essentially memorize the data to determine patterns, rather than learn the patterns to make classification predictions. In each of the above hyperparameter tests, it is concluded that when their values are increased – increased number of hidden layers, increased number of neurons in each hidden layer, increased number of iterations – the overall accuracy of the model, generally, increases. It also should be noted that when increasing these parameters, the run-time also increases, which aligns with the fact that the model complexity is rising and thus takes more time to traverse and back propagate through the hidden layers and neurons. Furthermore, the testing set of the model helps in understanding which versions of the hyperparameters result in a more accurate model. For example, when increasing the iterations from 150 to 250, the testing set accuracy value slightly falls, which indicates that even though there are significantly more iterations, that does not necessarily mean that these parameter values are optimal for this dataset.