Assignment 7
Trevor Lacoste

```
#Dropping non-relevant variables
data.drop(columns=['Neo Reference ID'], inplace=True)
data.drop(columns=['Name'], inplace=True)
data.drop(columns=['Close Approach Date'], inplace=True)
data.drop(columns=['Epoch Date Close Approach'], inplace=True)
data.drop(columns=['Orbit ID'], inplace=True)
data.drop(columns=['Orbiting Body'], inplace=True)
data.drop(columns=['Orbit Determination Date'], inplace=True)
data.drop(columns=['Equinox'], inplace=True)

#StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

After importing the data set, I first dropped some of the irrelevant variables, including the dates, names, and IDs.

After scaling the data and setting X to include all the remaining variables besides hazardous, I tested the default MLP classifier, which has one hidden layer with 100 neurons.

```
X = data.drop(['Hazardous'], axis=1)
X.info()
y = data['Hazardous']
```

This yielded an accuracy of about 84% but failed to identify almost all true positives. After adding more hidden layers and changing the number of neurons in each, I kept getting the same results with a training accuracy of around 84%.

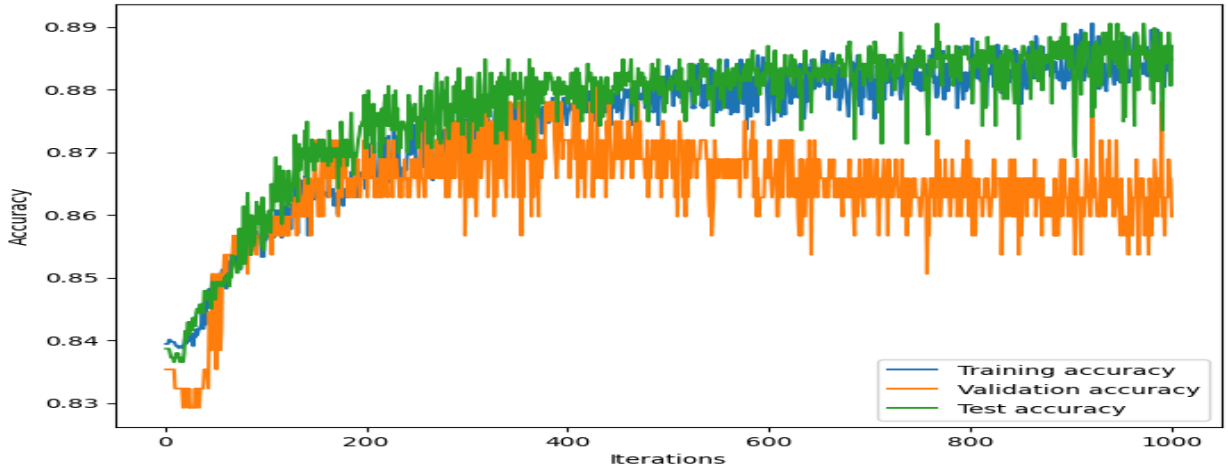
```
Overall Training Accuracy: 0.8465447154471545
Overall Test Accuracy: 0.84363894811656
```

In order to get a higher training accuracy, I decided to choose a subset of variables and see if that yielded better results. So, I picked variables that appeared to be good indicators of a hazardous asteroid and that weren't repeated with different units.

```
#Setting x to include only certain variables, and y as hazardous
X = data[['Absolute Magnitude', 'Est Dia in KM(min)', 'Semi Major Axis', 'Eccentricity', 'Semi Major Axis', 'Inclination', 'Perihelion Distance']]
X.info()
y = data['Hazardous']
```

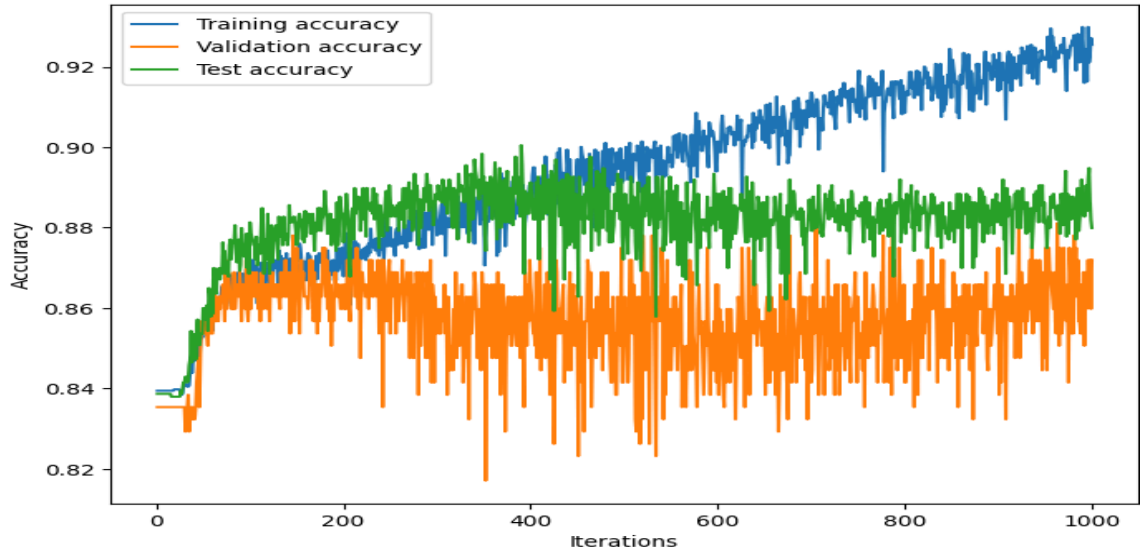
After making this change, I again tested the model with the default of one hidden layer with 100 neurons.

The result was this graph, which showed a higher accuracy without any changes to the hyperparameters.



After tweaking the number of hidden layers and neurons, I got to a model using three neurons with a training accuracy of 92.5% and a test accuracy of 87.9%. I decided this was better than the other models, even though the classifier with four hidden layers (100, 75, 50, 25) had a higher training accuracy of 97.5%. This is because the test accuracy was the same, so this model with four hidden layers seemed overfitted.

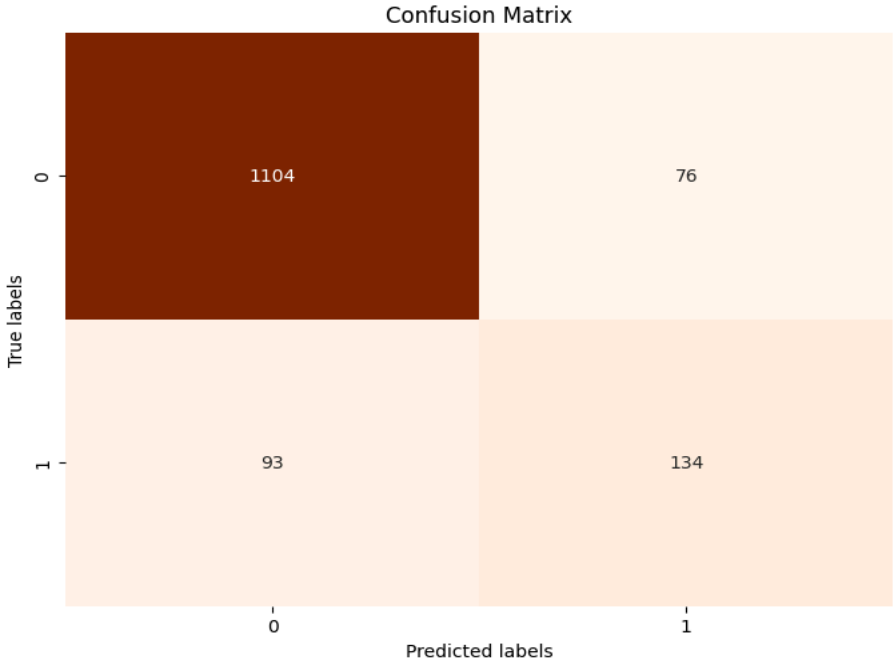
Final Model:




```
Overall Training Accuracy: 0.9254742547425474
Overall Test Accuracy: 0.8798862828713575
```

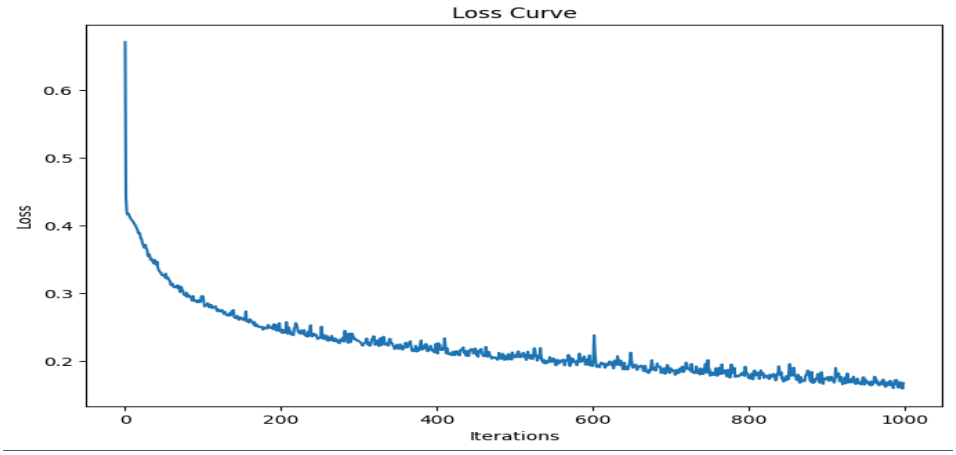
The above plot and accuracies are from an MLP classifier with three hidden layers with 70, 50, and 30 neurons.

Confusion Matrix:



The confusion matrix shows the model predicted 1104 true negatives and 134 true positives. It also had 76 false positives and 93 false negatives. So, it is very accurate at classifying true negatives, but could be better at finding true positives. This means the model can easily identify non-hazardous asteroids but is not as good at predicting hazardous asteroids.

The Loss Curve:



The loss curve converges towards 0 over time, showing that the model is minimizing loss through the training process, meaning it is improving over the iterations. Towards the end of iterations, it does start to flatten out, meaning that training has slowed down a bit, which is expected.

Conclusion:
Using all of the numerical variables provided did not provide an accurate model to predict whether or not an asteroid was hazardous. Using a subset of variables made it easier to find an MLP classification that was relatively accurate at identifying hazardous and non-hazardous asteroids.

Table of tested values

# Hidden Layers	# Neurons	Training & Test Accuracy
1	100	88%, 88.9%
2	50, 25	88.8%, 88.1%
3	50, 25, 10	88.3%, 88.9%
3	60, 30, 10	91.8%, 88.9%
2	10, 5	86.9%, 87.6%
2	75, 50	90.6%, 88.2%
3	75, 50, 25	91.6%, 88.8%
4	75, 50, 25, 10	93%, 88.3%
3	60, 30, 15	90.5%, 87.8%
3	70, 50, 30	92.5%, 87.9%
3	70, 40, 20	90.6%, 88.1%
4	100, 75, 50, 25	97.5%, 89%
1	150	88.3%, 88.6%