Task 1 Report

Naca Hitchman - 103072170

# Environment Setup

A screenshot of a computer

Description automatically generatedI started by downloading the 0.1 and P1 code bases, storing them in their own directories.

A screenshot of a computer

Description automatically generated

I then viewed the code of and ran the 0.1 stock\_prediction.py file to determine the dependencies that would need to be in the requirements.txt.

A screenshot of a computer program

Description automatically generated

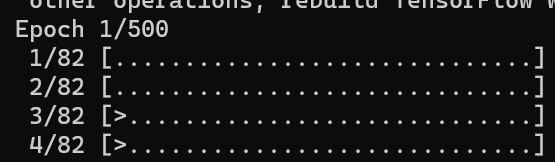
# Testing Code Bases

Running the 0.1 code base was relatively quick, going through only 25 epochs before opening the chart showing the actual and predicted TSLA stock price.

A screenshot of a computer screen

Description automatically generated

The initial run of the P1 code base, however, was set to iterate through 500 epochs, which would take significantly too long for the purposes of this task.



Thus, for this task I edited the parameters.py file to have the training only run through 10 epochs. I then ran the test.py file to build and view the results.

A screenshot of a computer

Description automatically generated

# Understanding the 0.1 Code Base

After running both the 0.1 and P1 code bases, I investigated the python code of the 0.1 code base to attempt to understand how it works.

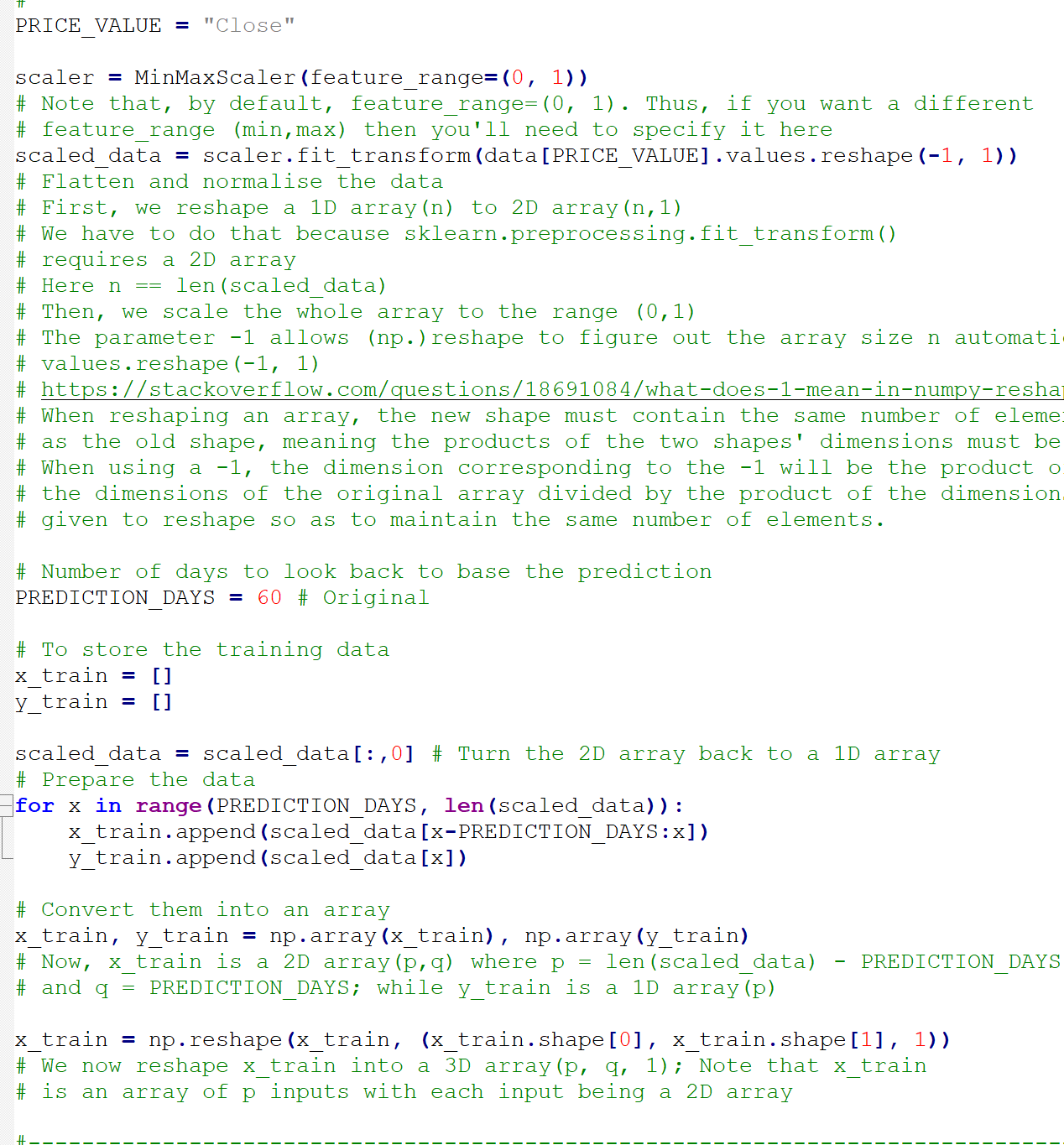
As is typical, it begins by importing the required dependencies and libraries to allow the system and machine learning to function.

A screenshot of a computer program

Description automatically generated Some global constants are then set including the source of the data, the stock name being predicted, and the dates to train the data on, before downloading the required data from the yfinance source.

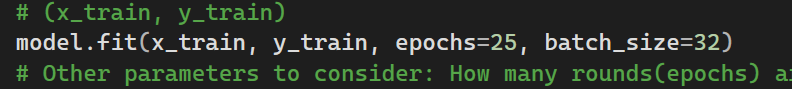
A close-up of a computer screen

Description automatically generated

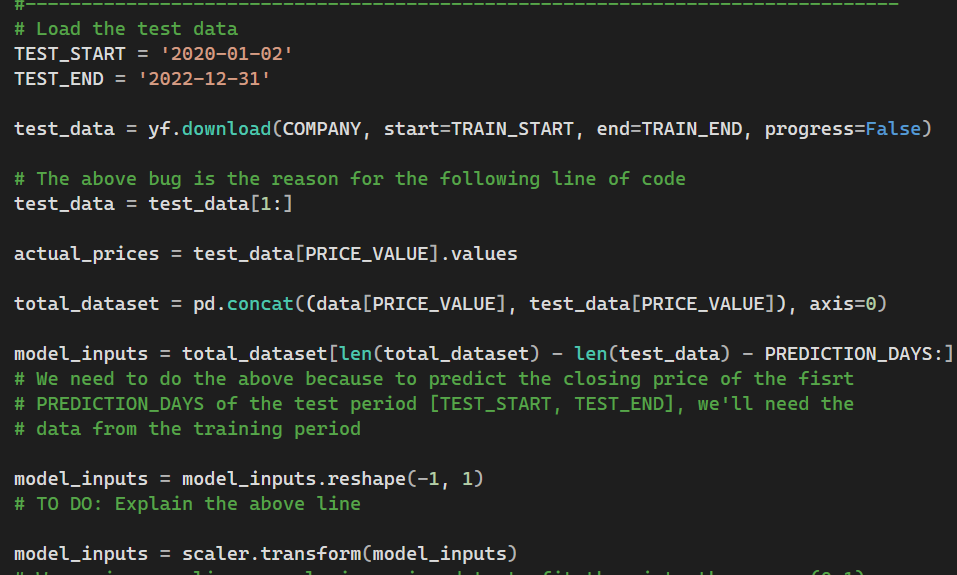
Following this, the data is prepared for usage, and the neural network model is created and its various properties and settings are defined. The model then gets compiled.



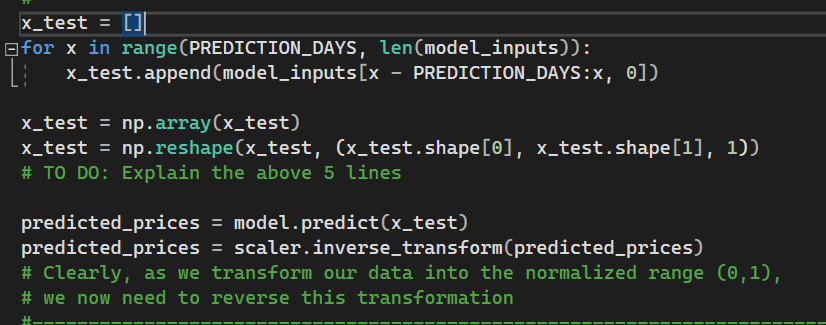
The model is then trained



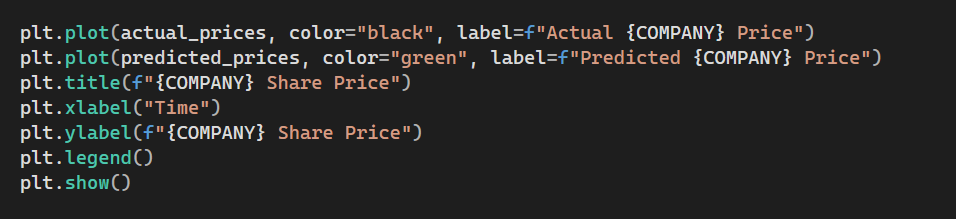
More recent test data is downloaded and formatted for usage to check the accuracy of the model



The relevant amount of days’ worth of data is converted to an array, before having the prediction run on it.



The prediction and actual prices are rendered onto a graph



Lastly, the next days’ price is also predicted and printed

