

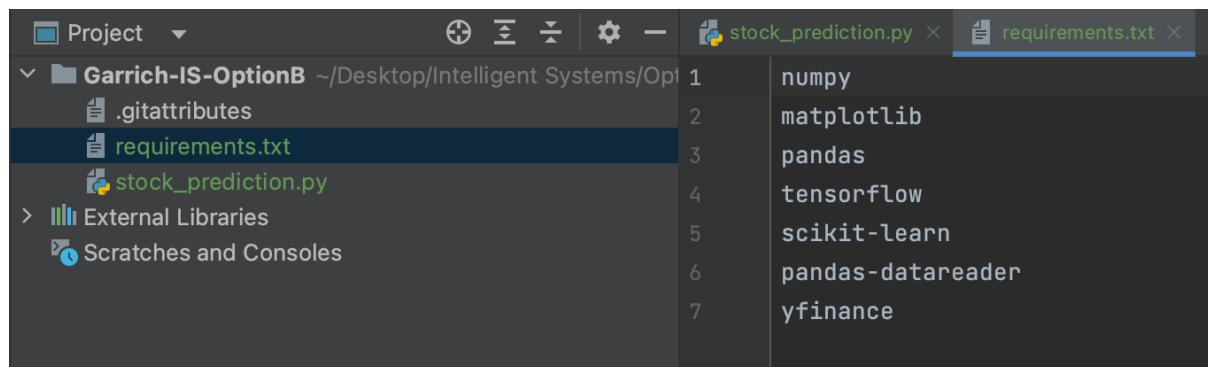
Task 1 Setup Report

Environment Setup

To get the environment setup, the following libraries needed to be downloaded.

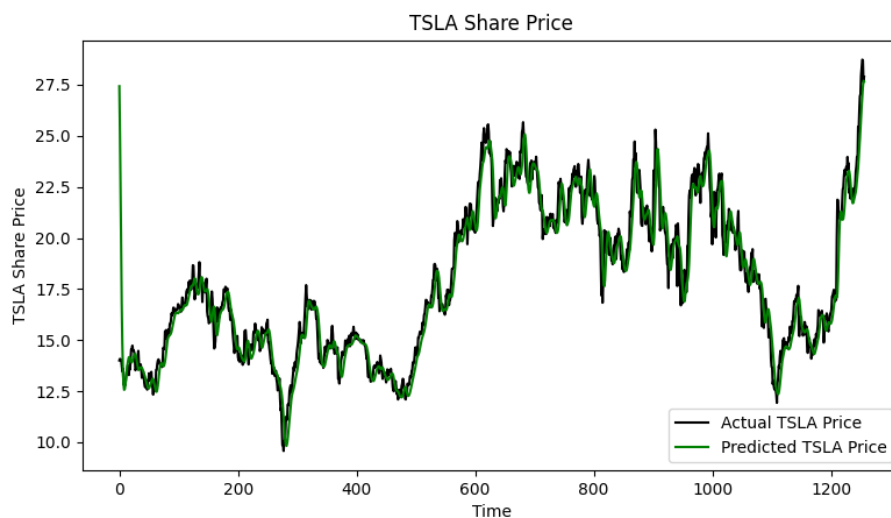
```
10 # Need to install the following:
11 # pip install numpy
12 # pip install matplotlib
13 # pip install pandas
14 # pip install tensorflow
15 # pip install scikit-learn
16 # pip install pandas-datareader
17 # pip install yfinance
```

To make it easier in the future to install these libraries and reduce the number of lines of code. I have created a “requirements.txt” text file which has the libraries inside, by doing this, in the future if someone wants to use the file, they only must type one line of code. “pip install -r requirements.txt”. If the IDE supports it, a prompt would be given whether the user wants to download the libraries from the requirements text file which completely removes the need for the extra line of code to be downloaded.



Testing v0.1

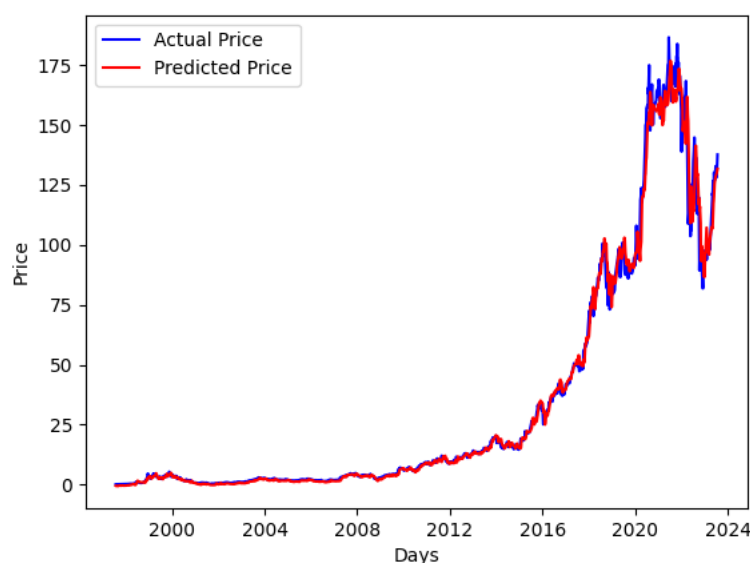
After getting the libraries installed, I ran the stock prediction program prepared by NeuralNine and it runs properly as needed, the following output was created.



Testing P1

Initially trying to run the script for P1 was a bit troublesome as the IDE “PyCharm” was not working well with tensorflow and not recognizing the keras library inside the script. Resulting in “tensorflow.keras” not working, I troubleshooted this problem and for the time being the only solution is to import tensorflow as tf then manually type in tf.keras.xxx whenever it is needed. Although inefficient this is the only method that is functional until another more efficient solution is found.

After getting the libraries to load properly for this, I trained the algorithm on 25 Epochs instead of the default 500 which was set at the beginning as that would just take way too long. Then I tested it and below is the output.



Summary of what v0.1 does

From my understanding, the program initially loads the required libraries. Followed by retrieving and loading data from what is currently set as yahoo finance and the chosen stock being TSLA for Tesla. The dates from which the training data starts, and ends is also specified (2015 – 2020). The value of the stock is also taken from the price it closed on at each date. The dataset is then ran through a MinMaxScaler which scales the closing prices into values between 0 and 1 (default setting). The data is then trained using the Sequential model on 60 days of past data. The current model has 3 LSTM layers and 1 dense layer for prediction. The model is then optimized using the adam optimizer and uses mean_squared_error for its loss parameter. The model repeats its training 25 times or 25 epochs. The trained data is then tested on data from 2020 to 2022. The results are then displayed on a line graph plot using matplotlib library. After that a prediction of what the next closing price for the following day will be.