

Attempt Summary – Senithu Perera - 104208468

Introduction

This is the report of the attempt made for v0.1 and P1, Screenshots were taken during the testing process to document the outputs and any issues encountered. These screenshots provide a visual summary of the testing process and the current state of the project.

Initial Code base v0.1

The initial code base v0.1 serves as the foundational structure for a stock price prediction model using a recurrent neural network (RNN), specifically focusing on a Long Short-Term Memory (LSTM) network in Python. The code is organized to facilitate the prediction of a stock's next-day closing price based on the last 60 days of historical data. The model's primary objective is to provide a basic introduction to machine learning concepts, rather than producing highly accurate predictions for investment purposes.

The code begins by importing essential Python libraries such as ``numpy`` for numerical operations, ``matplotlib`` for plotting, ``pandas`` for data manipulation, and TensorFlow/Keras for building the neural network. It utilizes ``pandas_datareader`` to fetch historical stock price data, which is then preprocessed by scaling the closing prices between 0 and 1 using ``MinMaxScaler`` from ``scikit-learn``.

The LSTM model is constructed using TensorFlow's Sequential API, with multiple LSTM layers interspersed with dropout layers to mitigate overfitting. A dense layer at the end produces the final prediction. The model is compiled with the Adam optimizer and mean squared error as the loss function, reflecting its regression task.

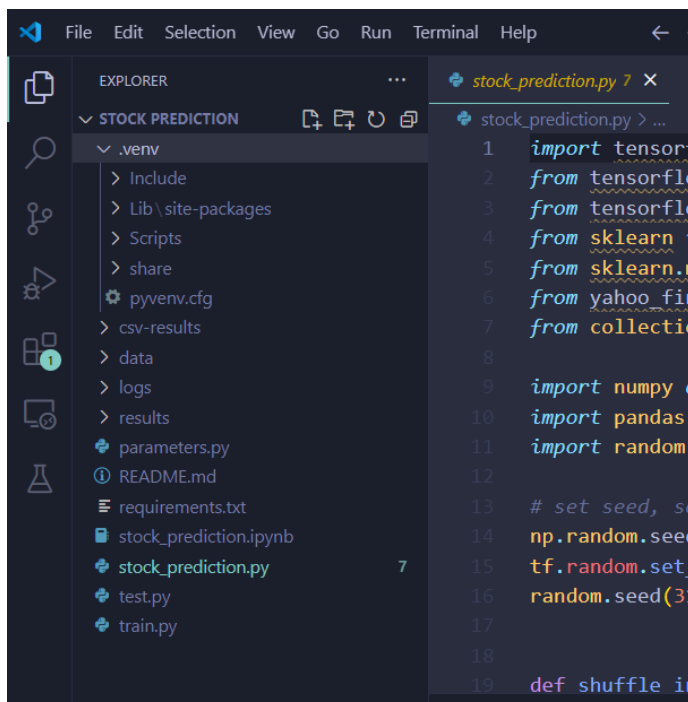
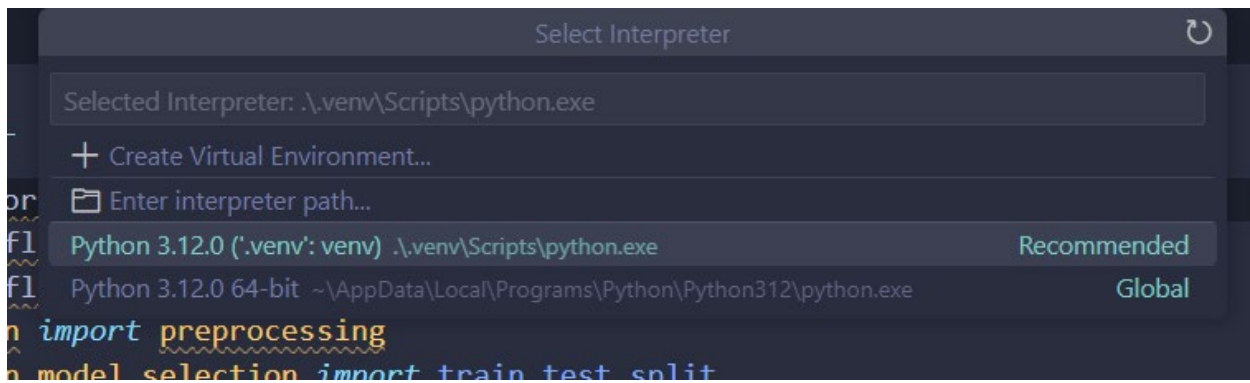
The code also includes functionality for evaluating the model's accuracy by comparing its predictions to actual historical prices. Despite its simplicity, the v0.1 code base offers a solid introduction to the concepts of RNNs, LSTMs, and time-series forecasting, laying the groundwork for more sophisticated models in future iterations.

Setting up Virtual Environment

On this stage the virtual environment was created in visual studio code using the following code in a new terminal

```
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction> python3 -m venv .venv
```

Then after terminating the terminal, the correct version of the python interpreter was selected to the virtual environment. If this is not selected the libraries will not be resolved into the code base from vs code



Creating the Requirements.txt

A requirements file was created to list all the necessary packages. This file is essential for setting up the environment consistently across different systems.

```
[notice] A new release of pip is available: 23.2.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction> pip3 install -r requirements.txt
Requirement already satisfied: scikit-learn in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 1)) (1.5.1)
Requirement already satisfied: tensorflow in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 2)) (2.17.0)
Requirement already satisfied: matplotlib in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 3)) (3.9.2)
Requirement already satisfied: numpy in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 4)) (1.26.4)
Requirement already satisfied: pandas in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 5)) (2.2.2)
Requirement already satisfied: yahoo_fin in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from -r requirements.txt (line 6)) (0.8.9.1)
Requirement already satisfied: scipy>=1.6.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from scikit-learn->-r requirements.txt (line 1)) (1.14.0)
Requirement already satisfied: joblib>=1.2.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from scikit-learn->-r requirements.txt (line 1)) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from scikit-learn->-r requirements.txt (line 1)) (3.5.0)

Requirement already satisfied: lxml>=2.1 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from pyquery->requests-html->yahoo_fin->-r requirements.txt (line 6)) (5.3.0)
Requirement already satisfied: cssselect>=1.2.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from pyquery->requests-html->yahoo_fin->-r requirements.txt (line 6)) (1.2.0)
Requirement already satisfied: zipp>=0.5 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from importlib-metadata>=1.4->pyppeteer>=0.0.14->requests-html->yahoo_fin->-r requirements.txt (line 6)) (3.20.0)
Requirement already satisfied: colorama in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from tqdm<5.0.0,>=4.42.1->pyppeteer>=0.0.14->requests-html->yahoo_fin->-r requirements.txt (line 6)) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.1.1 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from werkzeug>=1.0.1->tenso
rboard<2.18,>=2.17.0->tensorflow-intel==2.17.0->tensorflow->-r requirements.txt (line 2)) (2.1.5)
Requirement already satisfied: soupsieve>1.2 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from beautifulsoup4->bs4->requ
ests-html->yahoo_fin->-r requirements.txt (line 6)) (2.6)
Requirement already satisfied: markdown-it-py>=2.2.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from rich->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow->-r requirements.txt (line 2)) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from rich->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow->-r requirements.txt (line 2)) (2.18.0)
Requirement already satisfied: mdurl<=0.1 in d:\swinburne\year 2 - sem 2\intelligent systems\stock prediction\venv\lib\site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow->-r requirements.txt (line 2)) (0.1.2)

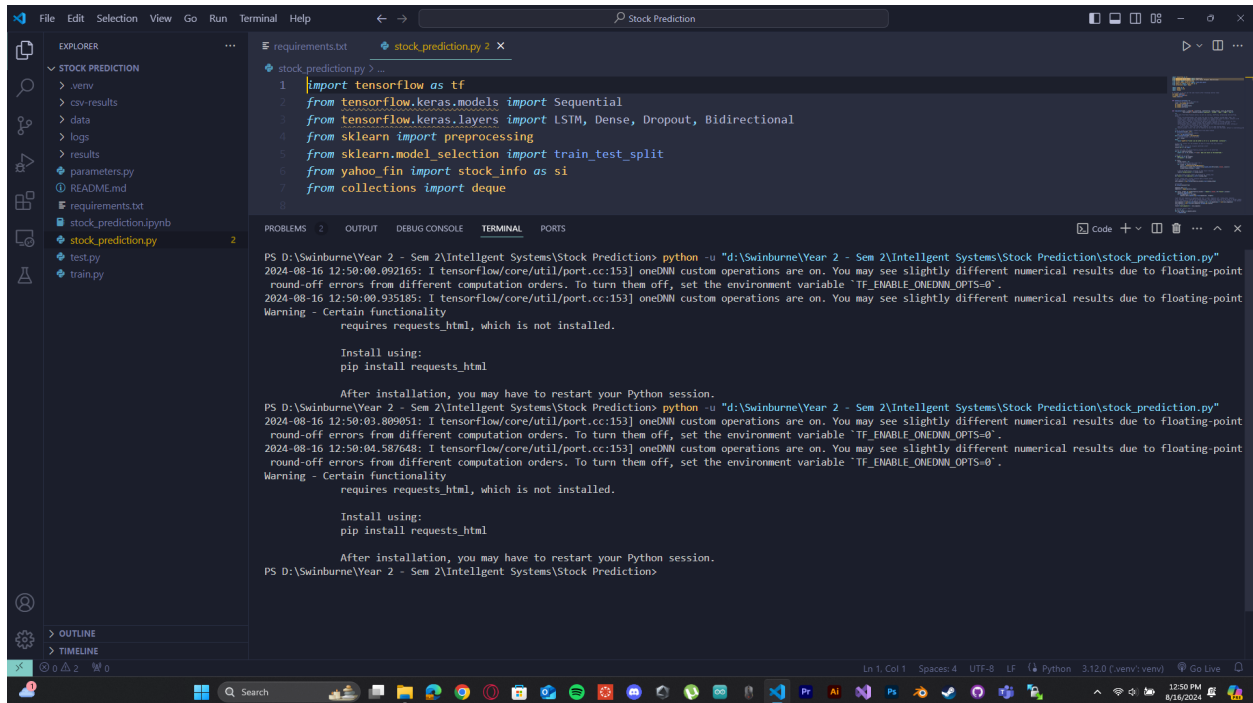
[notice] A new release of pip is available: 23.2.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction> 
```

When the code was ran without editing anything in the requirements file, an error comes up, this error can be solved by changing the ‘sklearn’ statement in the file to ‘scikit-learn’

```
requirements.txt X stock_prediction.py 1
requirements.txt
1 scikit-learn
2 tensorflow
3 matplotlib
4 numpy
5 pandas
6 yahoo_fin
```

Fixing the Environment Variable Error

An error was encountered when running the code as is, the error is referred in the following screen shot,



```
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction> python -u "d:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction\stock_prediction.py"
2024-08-16 12:50:00.092165: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2024-08-16 12:50:00.935185: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
Warning - Certain functionality
requires requests_html, which is not installed.

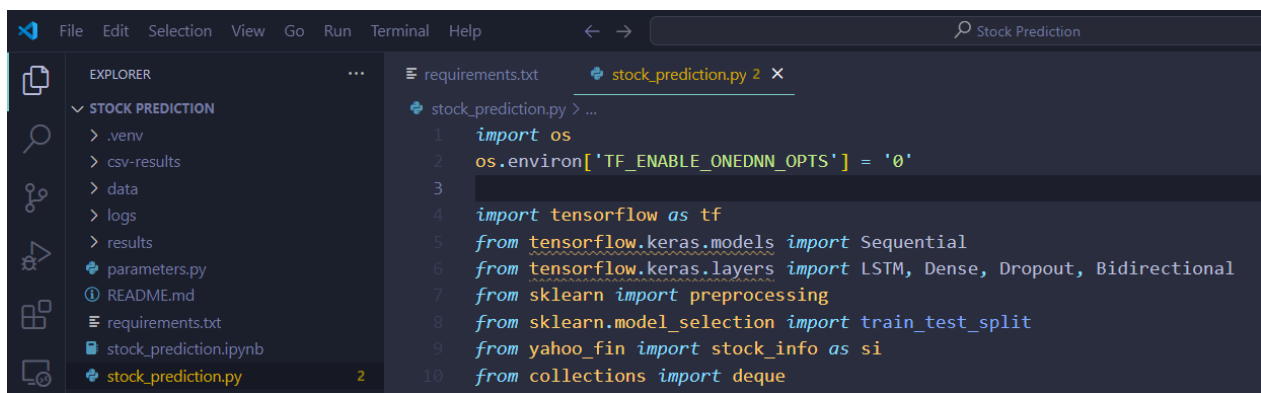
Install using:
pip install requests_html

After installation, you may have to restart your Python session.
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction> python -u "d:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction\stock_prediction.py"
2024-08-16 12:50:03.809051: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2024-08-16 12:50:04.587648: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
Warning - Certain functionality
requires requests_html, which is not installed.

Install using:
pip install requests_html

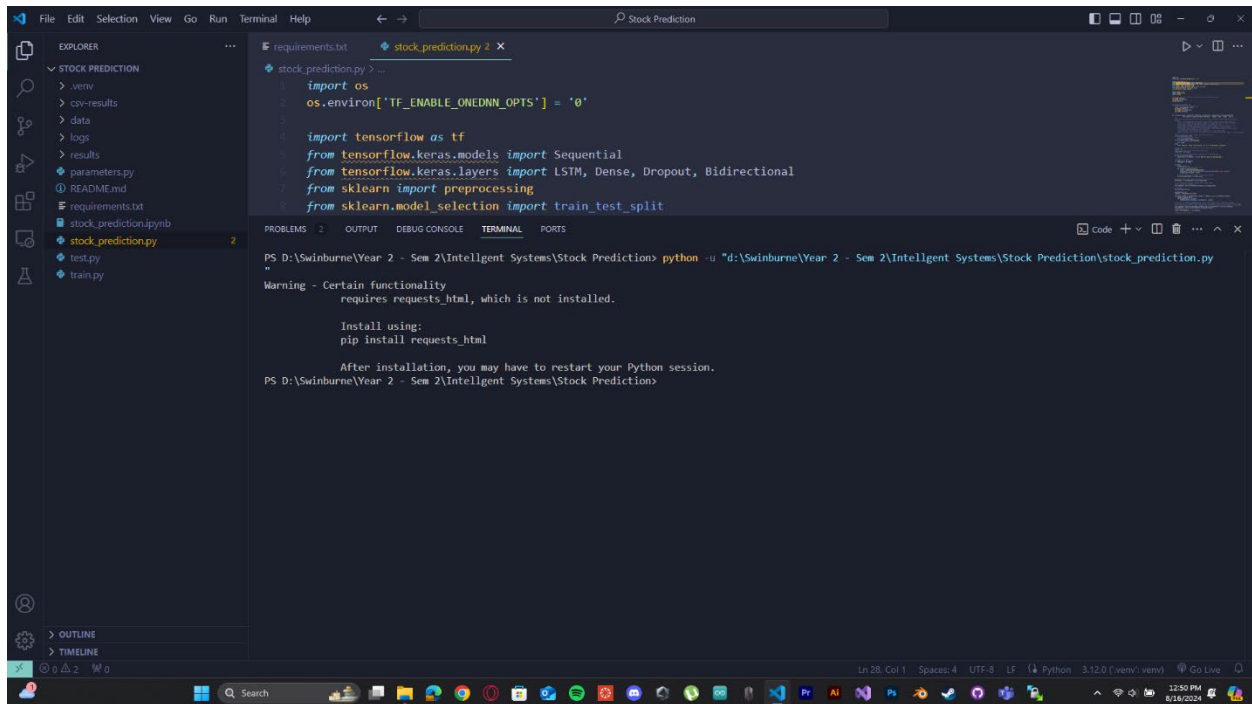
After installation, you may have to restart your Python session.
PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction>
```

this error can be resolved by adding By Importing the library “OS” we can change the environment variable in runtime



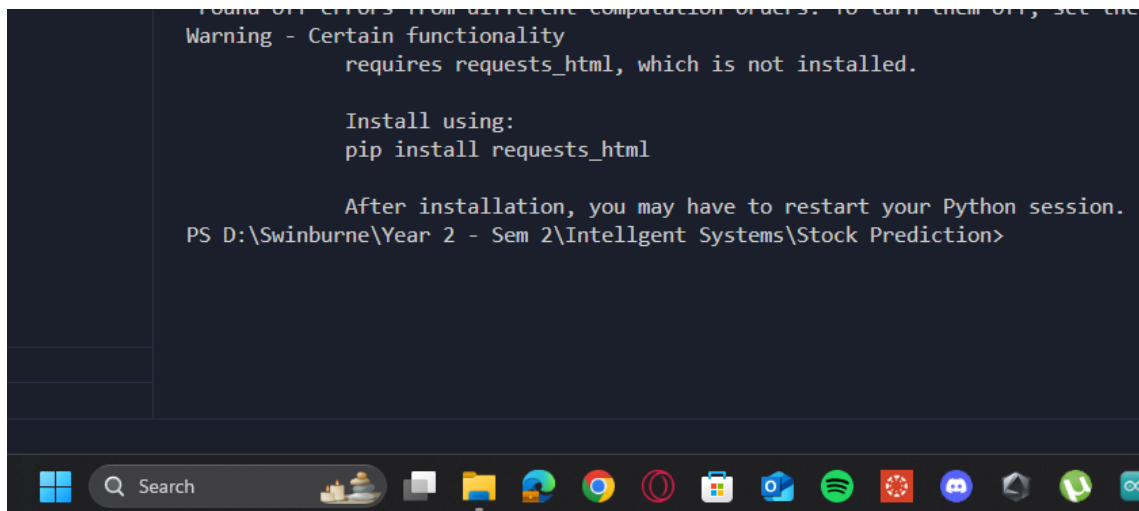
```
1 import os
2 os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'
3
4 import tensorflow as tf
5 from tensorflow.keras.models import Sequential
6 from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional
7 from sklearn import preprocessing
8 from sklearn.model_selection import train_test_split
9 from yahoo_finance import StockInfo as si
10 from collections import deque
```

Error fixed :-



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'STOCK PREDICTION' with files like 'requirements.txt', 'stock_prediction.py', 'test.py', and 'train.py'. The 'stock_prediction.py' file is open in the editor, showing Python code that imports 'os', 'tensorflow', 'keras', 'sklearn', and 'train_test_split'. The terminal at the bottom shows the command 'python -u "d:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction\stock_prediction.py"' being executed. The output of the command is a warning message: 'Warning - Certain functionality requires requests_html, which is not installed. Install using: pip install requests_html. After installation, you may have to restart your Python session.' The terminal prompt is 'PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction>'.

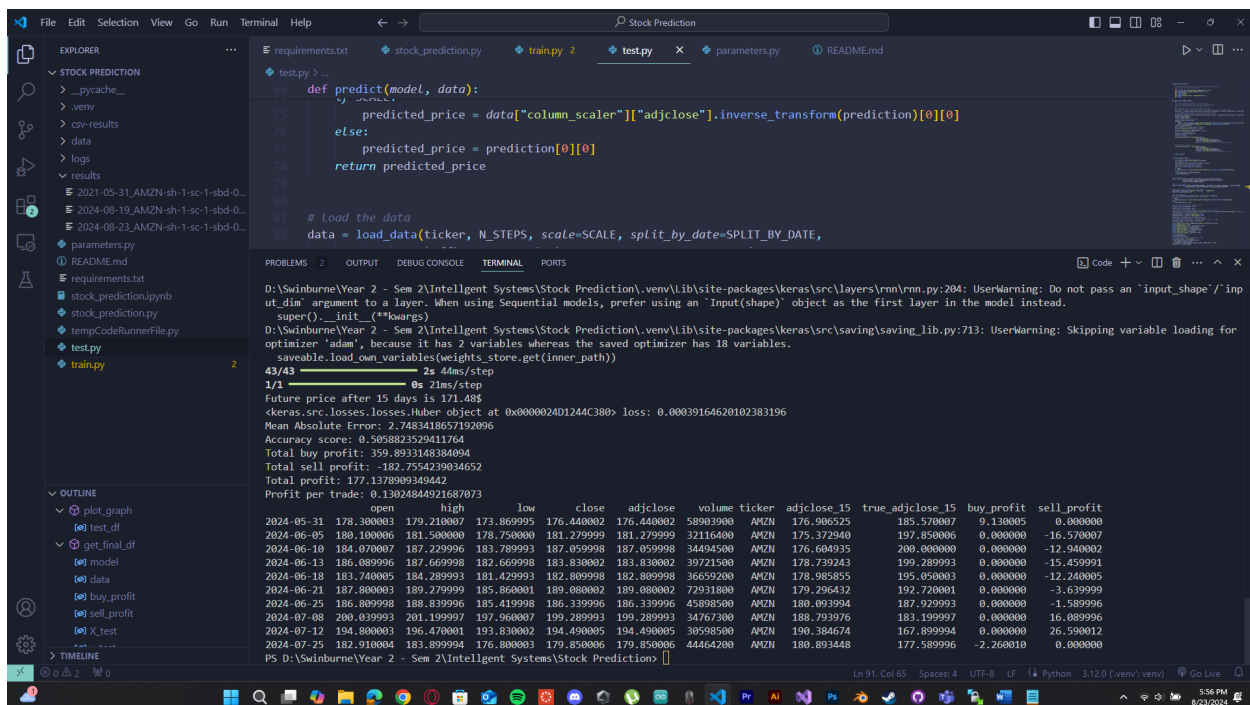
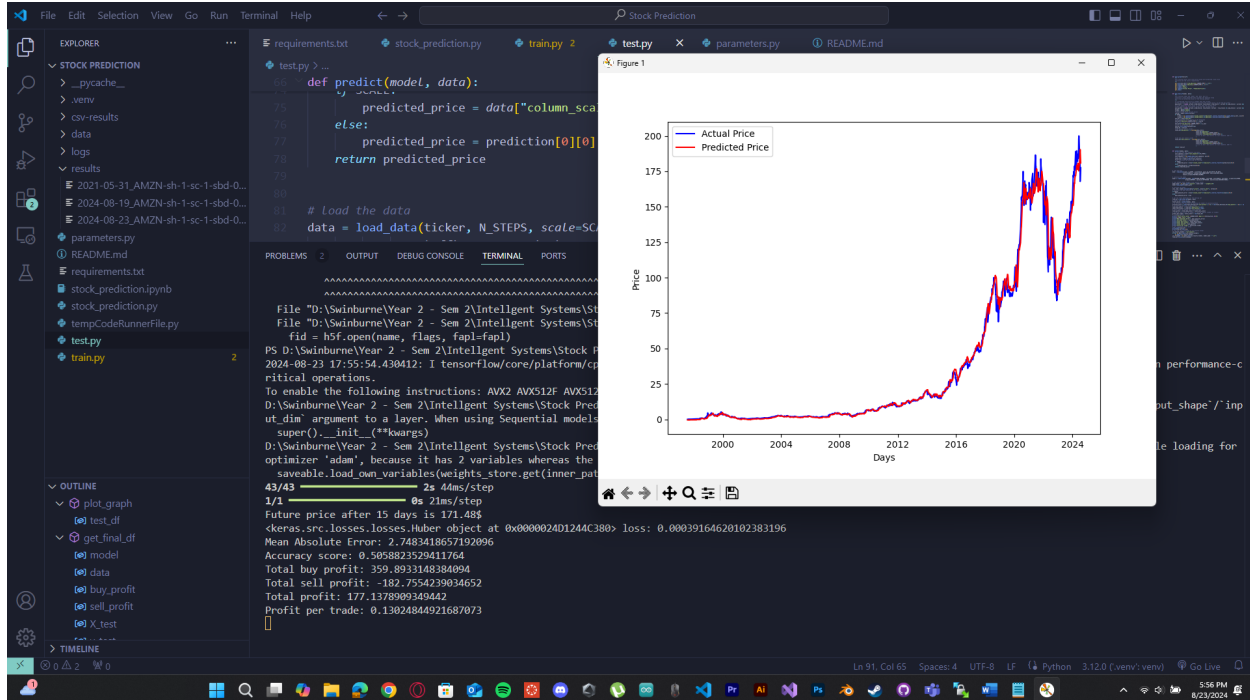
Fixing the error related to requests_html library



The screenshot shows a terminal window with the following text: 'Warning - Certain functionality requires requests_html, which is not installed. Install using: pip install requests_html. After installation, you may have to restart your Python session. PS D:\Swinburne\Year 2 - Sem 2\Intelligent Systems\Stock Prediction>'. The terminal is running on a Windows operating system, as indicated by the taskbar at the bottom.

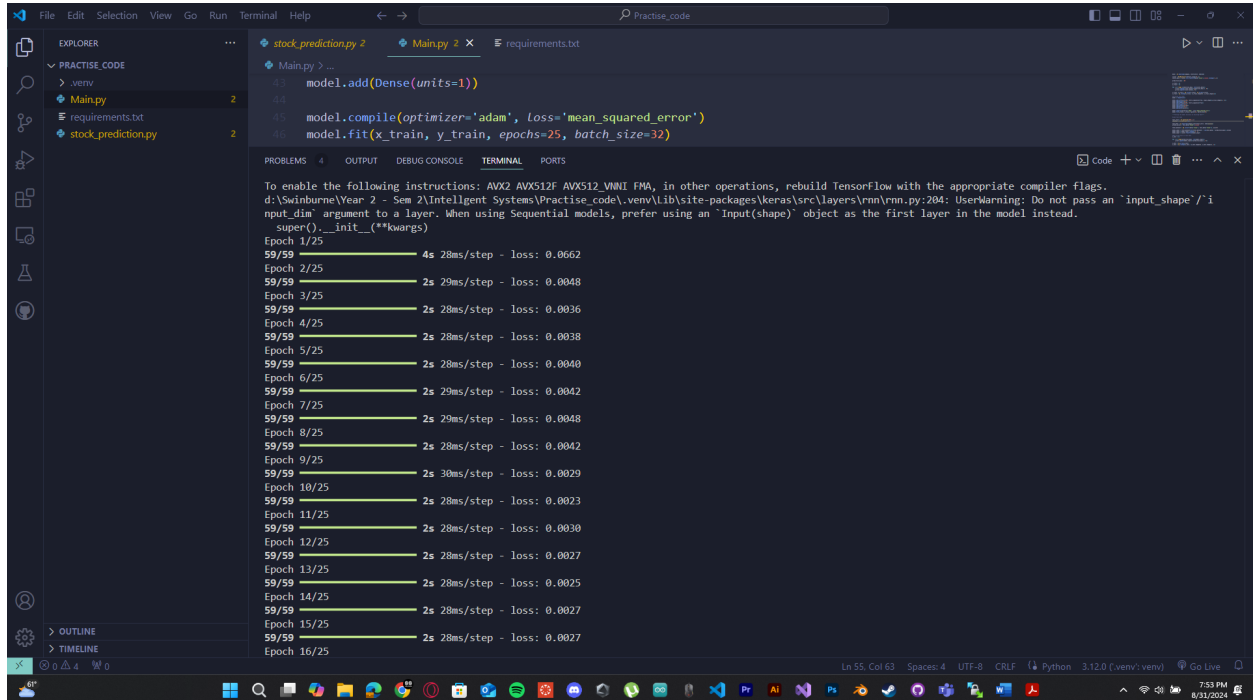
import requests_html // This will be added on top of the code

Even after restarting the python session this wouldn't work, the solution come down to adding an import statement with the library at the top



V 01 update code base Output

Training



```
File Edit Selection View Go Run Terminal Help  
Practise_code  
EXPLORER  
PRACTISE_CODE  
  .env  
  Main.py  
  requirements.txt  
  stock_prediction.py  
Main.py  
43 model.add(Dense(units=1))  
44  
45 model.compile(optimizer='adam', loss='mean_squared_error')  
46 model.fit(x_train, y_train, epochs=25, batch_size=32)  
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.  
d:\Swinsburne\Year 2 - Sem 2\Intelligent Systems\Practise_code\venv\Lib\site-packages\keras\src\layers\rnn\rnn.py:204: UserWarning: Do not pass an 'input_shape' / 'input_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.  
  super().__init__(**kwargs)  
Epoch 1/25  
59/59 4s 28ms/step - loss: 0.0662  
Epoch 2/25  
59/59 2s 29ms/step - loss: 0.0048  
Epoch 3/25  
59/59 2s 28ms/step - loss: 0.0036  
Epoch 4/25  
59/59 2s 28ms/step - loss: 0.0038  
Epoch 5/25  
59/59 2s 28ms/step - loss: 0.0040  
Epoch 6/25  
59/59 2s 29ms/step - loss: 0.0042  
Epoch 7/25  
59/59 2s 29ms/step - loss: 0.0048  
Epoch 8/25  
59/59 2s 28ms/step - loss: 0.0042  
Epoch 9/25  
59/59 2s 30ms/step - loss: 0.0029  
Epoch 10/25  
59/59 2s 28ms/step - loss: 0.0023  
Epoch 11/25  
59/59 2s 28ms/step - loss: 0.0030  
Epoch 12/25  
59/59 2s 28ms/step - loss: 0.0027  
Epoch 13/25  
59/59 2s 28ms/step - loss: 0.0025  
Epoch 14/25  
59/59 2s 28ms/step - loss: 0.0027  
Epoch 15/25  
59/59 2s 28ms/step - loss: 0.0027  
Epoch 16/25  
59/59 2s 28ms/step - loss: 0.0027  
Ln 55, Col 63 Spaces: 4 UTF-8 CRLF Python 3.12.0 (venv\venv) Go Live
```

Output

