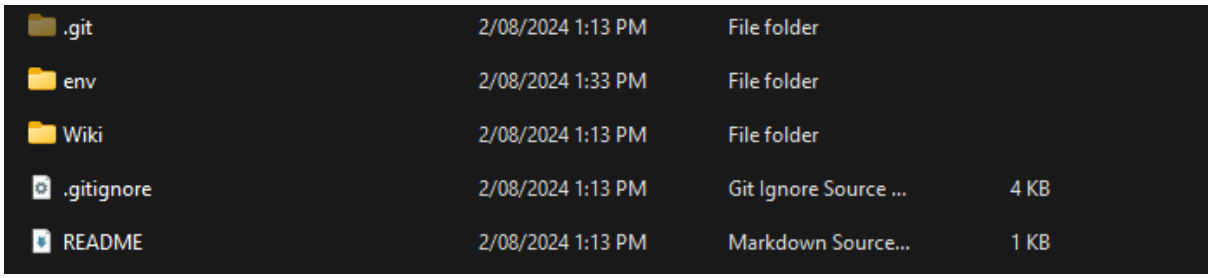
COS30018 Assignment B – Task 1

Aidan Grimmett: 103606838 – Friday 2:30 class

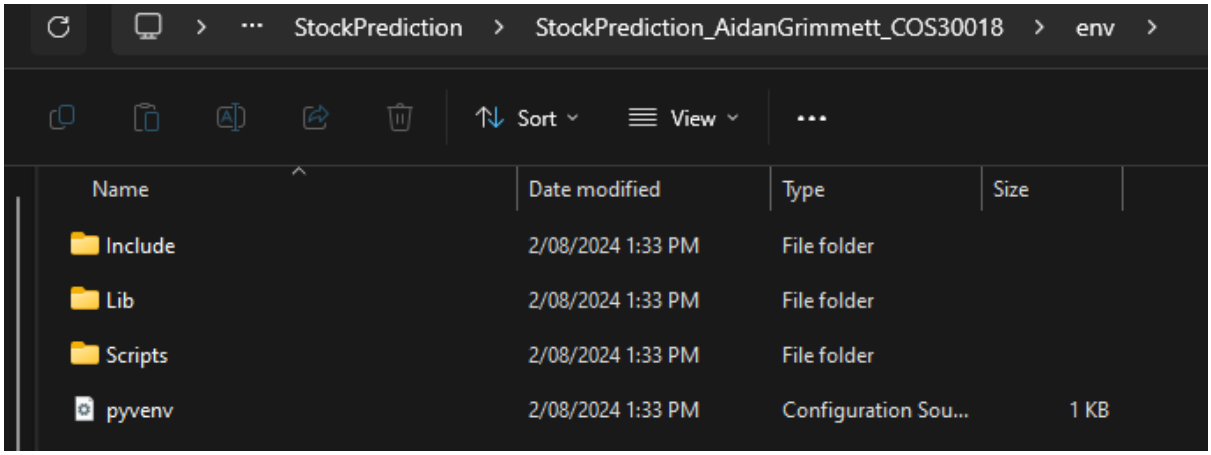
The first thing I did is create a new github repository, here is a link to the wiki page: https://github.com/AidanGrimmettSwin/StockPrediction_AidanGrimmett_COS30018/wiki

From there, I tried to create a virtual environment using venv as I had used it before.

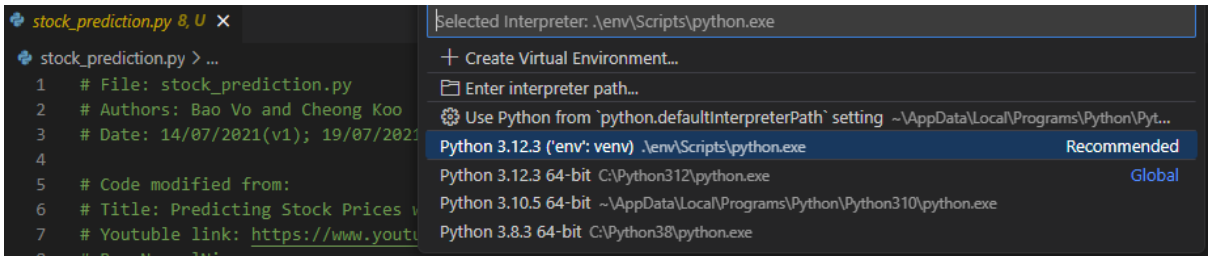
Creating the venv:



Inside the 'env' environment structure:



Then, I had to select the venv interpreter in VSC where I am running the code from (ctrl + shift + P to bring up this menu):



Installing numpy and showing that that is the only other package installed: (my global pip has lots of packages)

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\aidsg\Documents\Repos\StockPrediction\StockPrediction_AidanGrimmett_COS30018> pip install numpy
Collecting numpy
  Downloading numpy-2.0.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
     ---------------------------------------- 60.9/60.9 kB 1.1 MB/s eta 0:00:00
Downloading numpy-2.0.1-cp312-cp312-win_amd64.whl (16.3 MB)
     ---------------------------------------- 16.3/16.3 MB 3.7 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.0.1

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\aidsg\Documents\Repos\StockPrediction\StockPrediction_AidanGrimmett_COS30018> pip list
Package Version
------- -------
numpy   2.0.1
pip     24.0
```

To install packages/modules in the virtual environment, I ran this command:

`pip install numpy matplotlib pandas tensorflow scikit-learn pandas-datareader yfinance`

Then to create the requirements file for future installs:

Run `python -m pip freeze > requirements.txt`

requirements.txt

```
absl-py==2.1.0
astunparse==1.6.3
beautifulsoup4==4.12.3
certifi==2024.7.4
charset-normalizer==3.3.2
contourpy==1.2.1
cycler==0.12.1
flatbuffers==24.3.25
fonttools==4.53.1
frozendict==2.4.4
gast==0.6.0
google-pasta==0.2.0
grpcio==1.65.2
h5py==3.11.0
html5lib==1.1
idna==3.7
joblib==1.4.2
```

.... Continues

However, I was running into issues with running the code still. Certain packages were still not being found (tensorflow, distutils). So instead I installed anaconda and used that to set up a new environment like so:

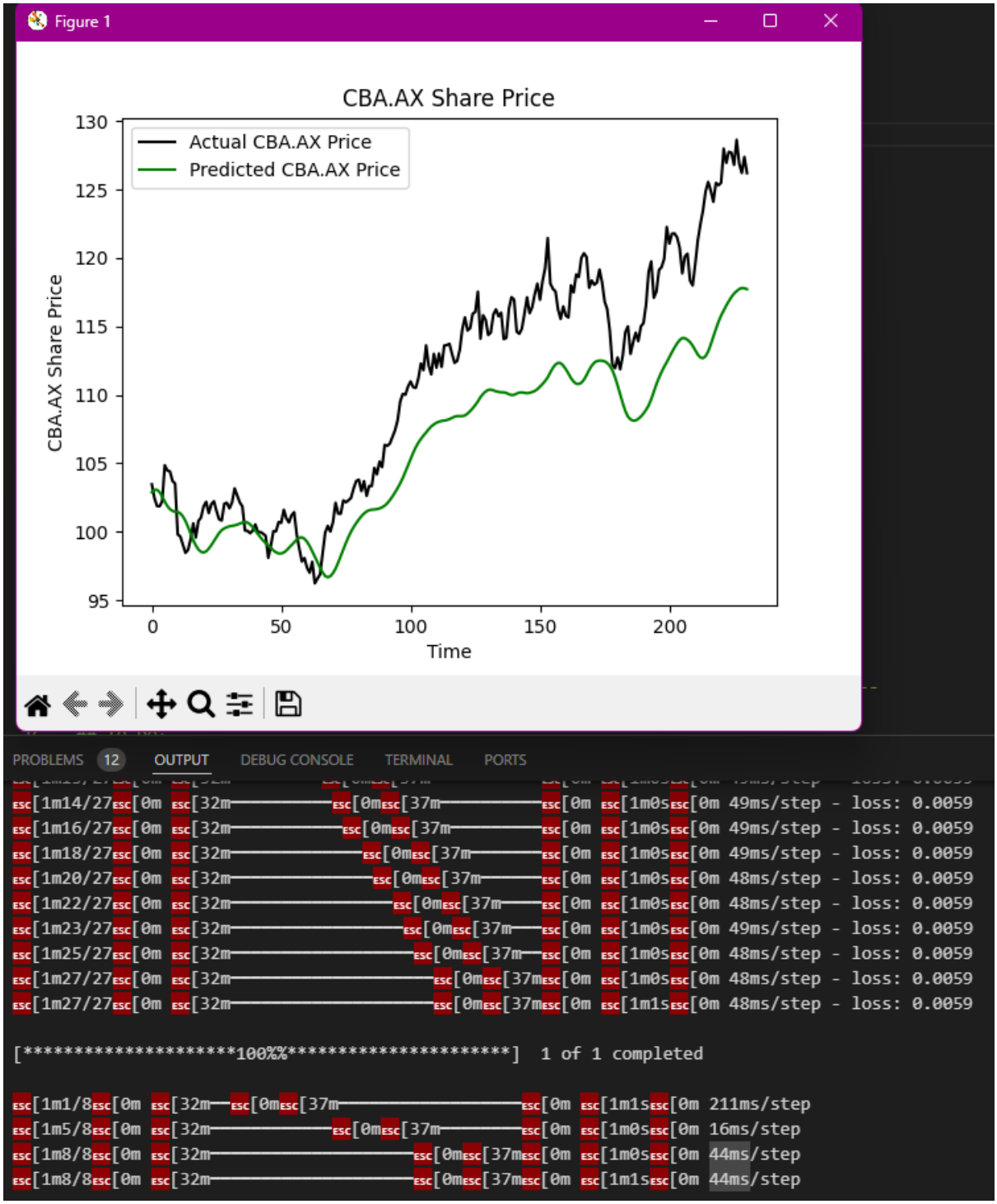
```
PS C:\Users\aidsg\Documents\Repos\StockPrediction\StockPrediction_AidanGrimmett_COS30018> conda create --name StockPredict1 -c anaconda python=3.10
Channels:
 - anaconda
 - defaults
Platform: win-64
```

Then I installed the required packages using the requirements.txt file, `pip install -r requirements.txt`. I had to be careful to do this inside the anaconda command prompt, trying to use conda commands through the terminal inside visual studio code was not working.

Now all of the packages were being imported correctly, but I ran into a missing charset issue, so I added these two lines to the top of the code to enforce a utf-8 encoding which fixed that issue.

```
# Can't run without specifying output encoding
import sys
sys.stdout.reconfigure(encoding='utf-8')
```

From there, the v0.1 code ran properly and I was able to very roughly predict the stock price as shown:



P1 task

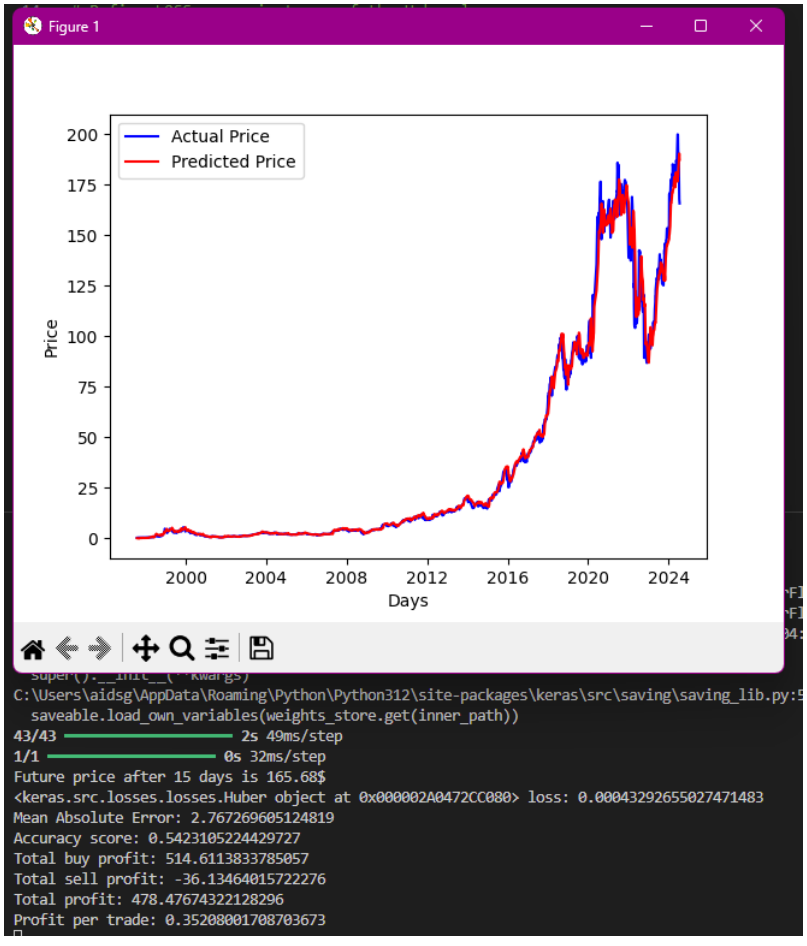
For the P1 code, I did set up a fresh conda environment as there was one different package (yahoo-finance vs yfinance in v0.1), and I thought I was getting some incompatibilities but it may have been another issue. There was one issue with the code which I fixed, I had to change what was appended to the model name from `model_name + ".h5"` to `model_name + ".weights.h5"`, as it wasn't recognising the file names.

```python
# some tensorflow callbacks
checkpointer = ModelCheckpoint(os.path.join("results", model_name + ".weights.h5"), save_weights_only=True, save_best_only=True, verbose=1)
tensorboard = TensorBoard(log_dir=os.path.join("logs", model_name))
# train the model and save the weights whenever we see
# a new optimal model using ModelCheckpoint
history = model.fit(data["X_train"], data["y_train"],
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_data=(data["X_test"], data["y_test"]),
                    callbacks=[checkpointer, tensorboard],
                    verbose=1)
```

Once that was changed, I was able to train the model as shown here:

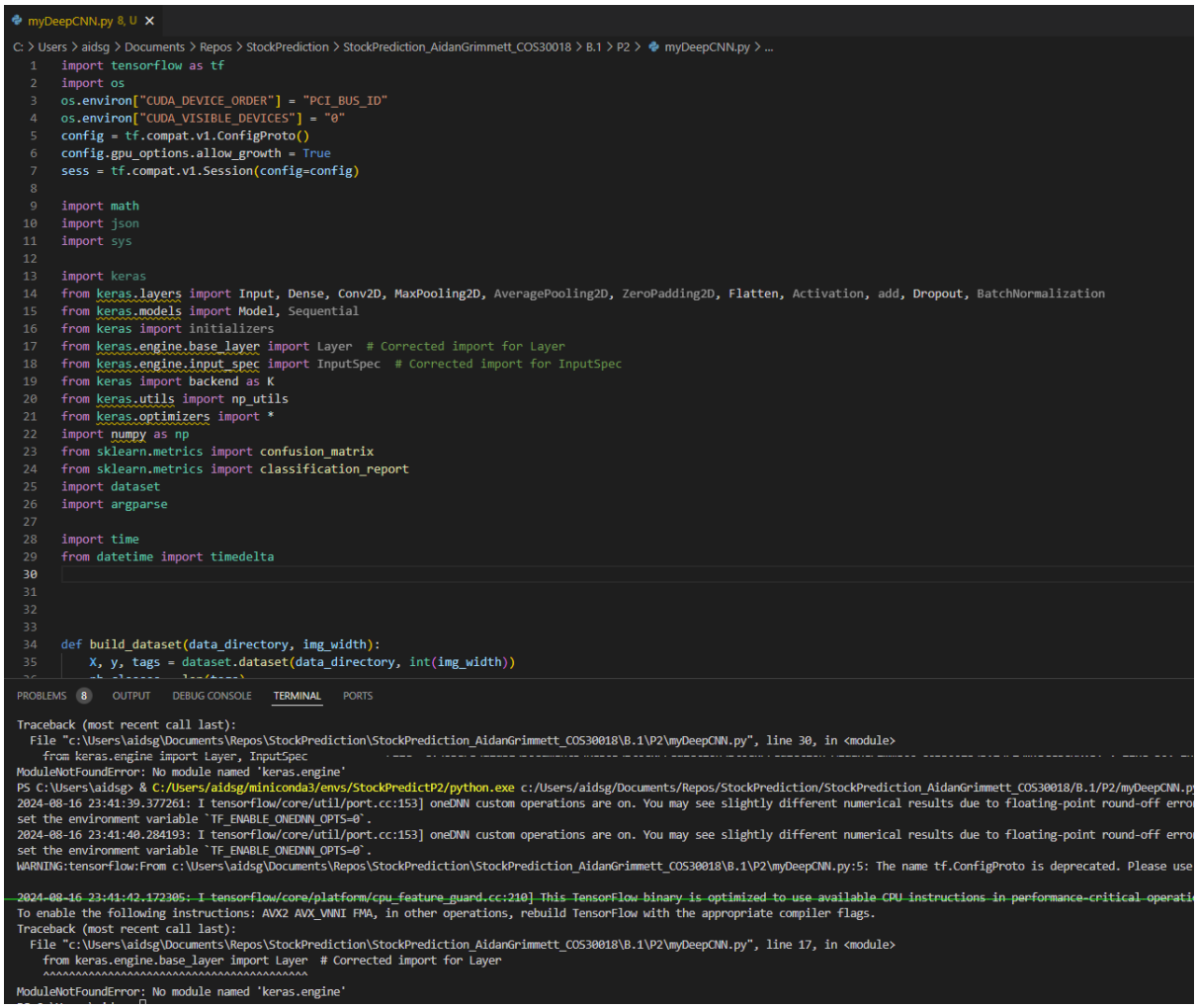
```
Epoch 1: val_loss improved from inf to 0.00050, saving model to results\2024-08-16_AMZN-sh-1-sc-1-sbd-0-huber_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.weights.h5
85/85 ━━━━━━━━━━━━━━━━━━━━ 27s 299ms/step - loss: 0.0066 - mean_absolute_error: 0.0500 - val_loss: 4.9782e-04 - val_mean_absolute_error: 0.0143
Epoch 2/5
85/85 ━━━━━━━━━━━━━━━━━━━━ 0s 278ms/step - loss: 7.8091e-04 - mean_absolute_error: 0.0196
Epoch 2: val_loss improved from 0.00050 to 0.00047, saving model to results\2024-08-16_AMZN-sh-1-sc-1-sbd-0-huber_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.weights.h5
85/85 ━━━━━━━━━━━━━━━━━━━━ 26s 302ms/step - loss: 7.8075e-04 - mean_absolute_error: 0.0196 - val_loss: 4.6796e-04 - val_mean_absolute_error: 0.0141
Epoch 3/5
85/85 ━━━━━━━━━━━━━━━━━━━━ 0s 281ms/step - loss: 6.9940e-04 - mean_absolute_error: 0.0183
Epoch 3: val_loss improved from 0.00047 to 0.00043, saving model to results\2024-08-16_AMZN-sh-1-sc-1-sbd-0-huber_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.weights.h5
85/85 ━━━━━━━━━━━━━━━━━━━━ 26s 309ms/step - loss: 6.9916e-04 - mean_absolute_error: 0.0183 - val_loss: 4.3293e-04 - val_mean_absolute_error: 0.0135
Epoch 4/5
85/85 ━━━━━━━━━━━━━━━━━━━━ 0s 285ms/step - loss: 7.4192e-04 - mean_absolute_error: 0.0195
Epoch 4: val_loss did not improve from 0.00043
85/85 ━━━━━━━━━━━━━━━━━━━━ 27s 313ms/step - loss: 7.4222e-04 - mean_absolute_error: 0.0195 - val_loss: 5.9813e-04 - val_mean_absolute_error: 0.0188
Epoch 5/5
85/85 ━━━━━━━━━━━━━━━━━━━━ 0s 265ms/step - loss: 6.8029e-04 - mean_absolute_error: 0.0192
Epoch 5: val_loss did not improve from 0.00043
85/85 ━━━━━━━━━━━━━━━━━━━━ 24s 287ms/step - loss: 6.8058e-04 - mean_absolute_error: 0.0192 - val_loss: 9.6708e-04 - val_mean_absolute_error: 0.0233
```

Then, I was able to run the test script to produce this stock prediction. It is significantly more accurate than the prediction created by the v0.1 code, and I only trained for 5 epochs, where it is common for training algorithms to run for dozens if not hundreds of epochs.



P2 Bonus Task

I also downloaded and attempted to run the P2 project. However, it seems that the project is missing its requirements file so I had to manually attempt to install all the required modules with `pip install numpy tensorflow keras scikit-learn`. The code seems to rely on deprecated versions of tensorflow however, and I did not allow enough time to modify the script and update it to work with current versions of tensorflow (I did try and downgrade my tensorflow to version 1.15 but pip was unable to locate any versions less than 2.16).



This was my attempt at running the code, after making some modifications to the import statements to try and help some of the problems.