

COS30018 – Intelligent Systems – Task B.1 Report

Ryan Chessum 102564760

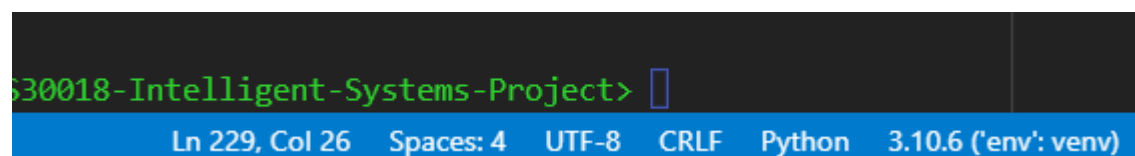
In this project I will be working on a stock price prediction system that utilises machine learning.

1.1 Setting up the project

The first step I took in this project was to set up the python virtual environment I would be using.

csv-results	19/08/2022 8:39 AM	File folder	
data	18/08/2022 11:54 PM	File folder	
env	18/08/2022 11:30 PM	File folder	
logs	18/08/2022 11:54 PM	File folder	
P1	19/08/2022 8:37 AM	File folder	
P2	18/08/2022 11:49 PM	File folder	
results	19/08/2022 8:37 AM	File folder	
v0.1	18/08/2022 11:32 PM	File folder	
.gitignore	18/08/2022 10:17 PM	Git Ignore Source ...	1 KB
README	18/08/2022 6:55 PM	Markdown Source...	1 KB
requirements	18/08/2022 11:28 PM	Text Document	1 KB

The environment is set up in the project folder. When I open the project in VS code it knows automatically to use the virtual environment instead of the version of python I already have installed on my machine.



env/ is also included in the .gitignore files as we don't need the virtual environment to be stored in the repository.

I also set up a requirements.txt file to easily install of the libraries needed for the P1 and v0.1 projects. Inside is a list of the pip arguments for all the dependencies.

```
requirements.txt
1  numpy
2  matplotlib
3  pandas
4  tensorflow
5  scikit-learn
6  pandas-datareader
7  yahoo_fin
8  sklearn
9  |
```

If we need to use the project on a separate machine, we can now create a new virtual environment and use the command *"pip install requirements.txt"* to install all the dependant libraries quickly and easily.

1.2 Testing the code

After setting up the environment we can now test the code.

Starting with v0.1 I simply ran the file and waited for it to finish running.

V0.1 is a simple stock price predictor. It first reads stock price data for a specified company off of yahoo using pandas data reader. Next it prepares the data for the neural network by scaling it down. We then add our data to 2 training sets. One of these data sets holds data for the last 60 days of stock prices while the second holds the closing price of the stock on the day after those 60 days which acts as our "prediction" dataset. We then convert these lists into arrays so they can be used in the neural network.

Next the neural network is defined it has 7 layers 3 lstm layers, 3 dropout layers and a dense layer for the next day prediction. The first lstm layer also acts as an input layer.

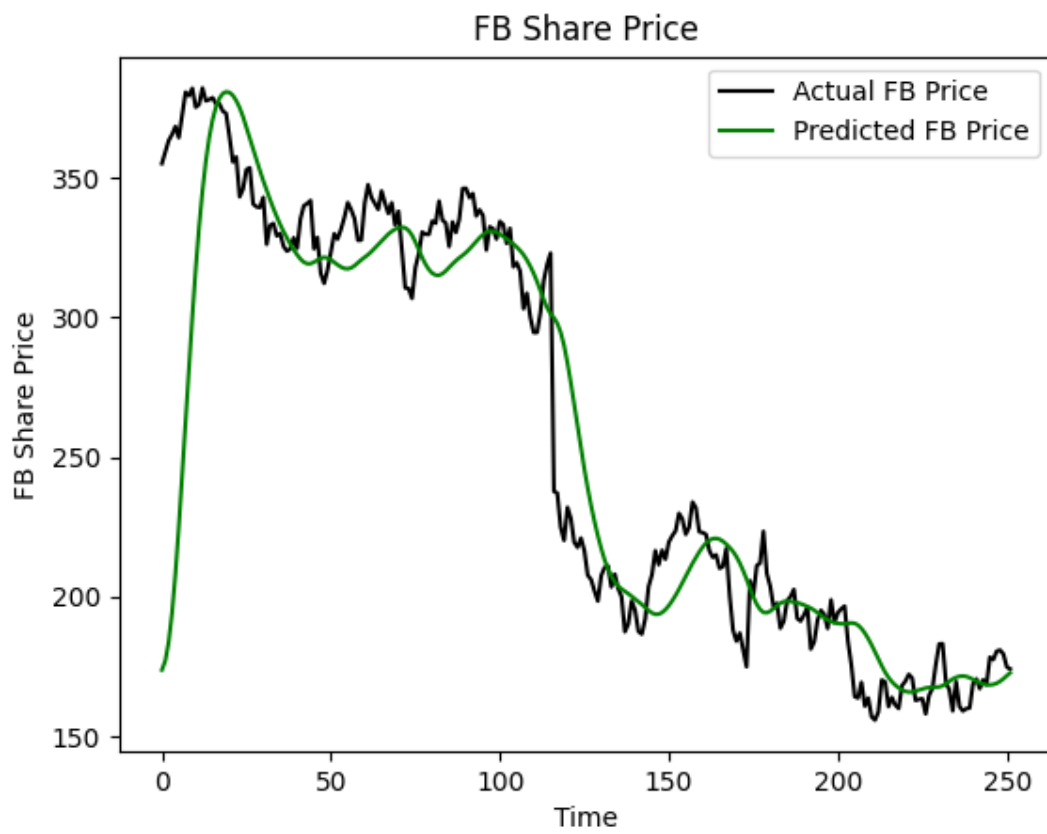
Once the neural network is compiled, we can train it on our data. Using the fit function passing in the data and setting epochs to 25 and batch size to 32.

Once the model has been trained. We test it on more data that it hasn't seen before to check its accuracy by grabbing more data off the web. The data is prepared for use similar to the training data with scaling and arrays. This time we only need 1 array as the model is making the predictions.

Then using the new data, the model makes predictions and we revert the data back to the original scale. The program then plots the data so us humans can interpret it more easily.

Finally, the model predicts the next close price using the real closing price data.

After running the code I got these results.



```
4/1 [Prediction: [[174.35242]]
```

P1 is a more complicated model than v0.1. After training the model over 500 epochs and running it I got these results.

