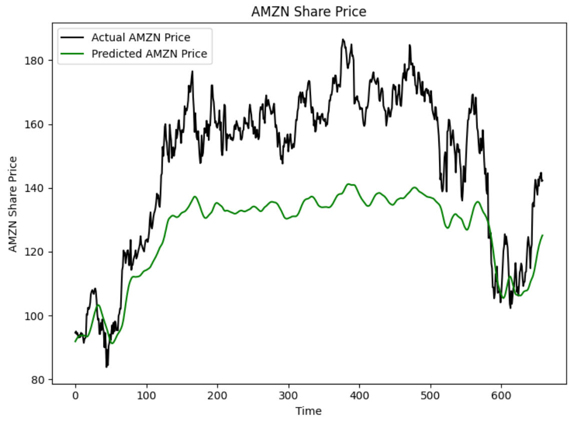
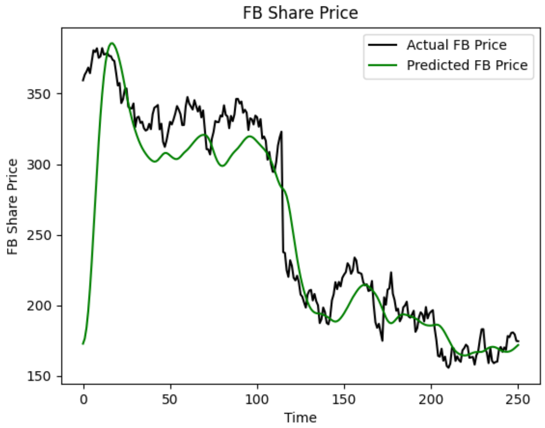
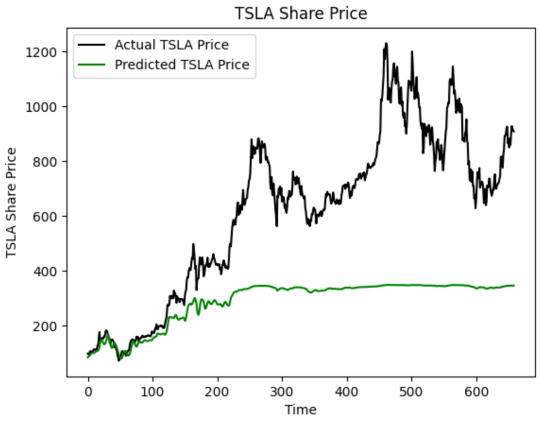
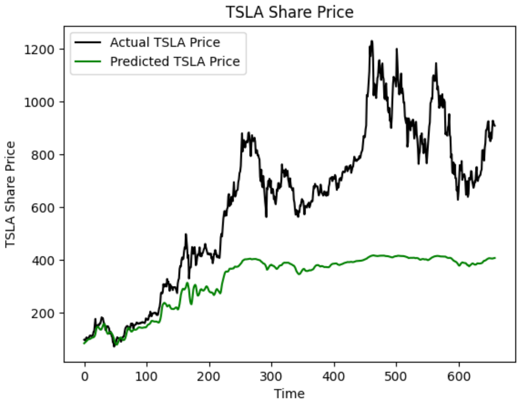
I first downloaded the two stock prediction files and having created separate files for the 2 versions, I used the requirements.txt with "pip install -r path/requirements.txt" and this allowed me to install the packages into a virtual environment.

I tested v0.1 and ended up with a couple results for FB, AMZN and TSLA



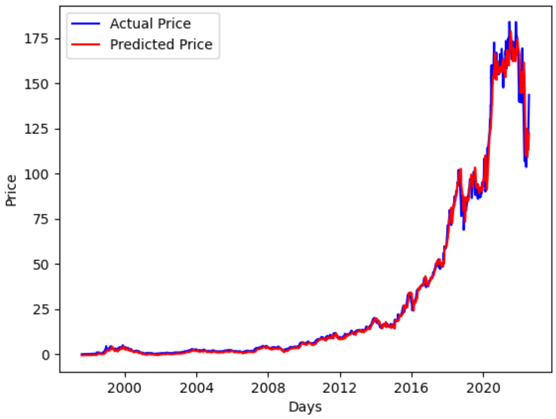


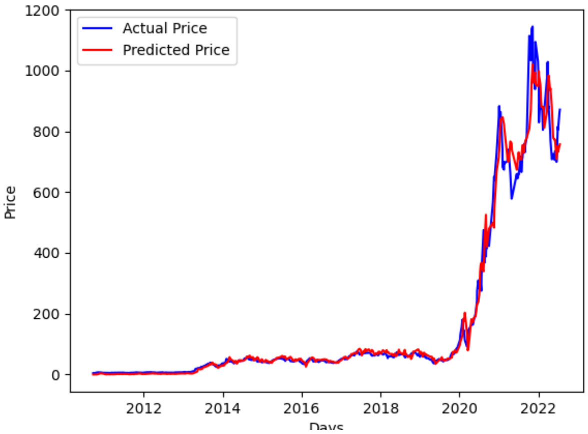




These results didn't take too long as the epoch was small, only being split into 7 batches and the time it took for the machine to load these datasets was relatively fast. The predictions are quite far off from what the actual prices are, TSLA seems to be the worst, but after changing the prediction days to 100 the model loaded looked a little more accurate.

P1 however took a while, the epoch was a much larger amount, being split into 500 separate batches and taking around 18 seconds per step, so training and loading one chart took 2 and a half hours. However the prediction looked to be quite accurate with the results of the prediction to actual price of AMZN and TSLA.





Summary of v0.1 code that required explaining:

The code considers the data of a stock and uses the data of the start date and end date inputted to read, this is taken from yahoo finance, after, it will take the last “specified” number of days to look back at and predict the stock price of the next day and compile that data into a visible model.

The code below takes the array and reshapes it without changing the data, here It takes the model_inputs and uses -1, this allows NumPy to calculate the dimensions for you if they are unknown, into a 2D array.

```
model_inputs = model_inputs.reshape(-1, 1)
```

In the code below x_test is a list and using a foreach loop in the range of PREDICTION_DAYS to the length in the model_inputs, add model_inputs of x minus prediction days up to x onto the x_test list and then using np.array turn x_test into an array and then is reshaped using the array dimensions so it has the same format.

```
x_test = []
for x in range(PREDICTION_DAYS, len(model_inputs)):
    x_test.append(model_inputs[x - PREDICTION_DAYS:x, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```