Problem 2: Recurrent Neural Networks for Regression

You are provided with a dataset for stock price prediction for 5 years with one sample per day (q2_dataset.py). Create a Recurrent Neural Network using the machine learning platform of your choice (PyTorch, Tensorflow, or Keras) to predict the next day opening price using the past 3 days *Open*, *High*, and *Low* prices and *volume*. Therefore, each sample will have (4*3 =) 12 features.

Follow the following steps and check the comments in the provided files and follow them:

1. In your train_RNN.py file: Before any preprocessing, create the dataset by using the latest 3 days as the features and the next day's opening price as the target. ***Randomize*** the created data and split it into 70% training and 30% testing and save it to '**train_data_RNN.csv**' and '**test_data_RNN.csv**' in the **data** directory respectively. Keep this code in your file but *comment it out*!
2. Populate the file **train_RNN.py** so that it reads your train_data_RNN.csv, preprocesses the data, and trains your RNN network. After training, the file should save your model with the name **'YOUR_ID_RNN_model'** in the **models** directory. You can use any extension you want as you will load your own model. Note that we will check that the test data was not used at ANY point in this file after saving it.
3. Populate the file **test_RNN.py** so that it reads your test_data_RNN.csv and runs the prediction model. Print the loss on your test data and show a plot of the true and predicted values (use appropriate labels for the axes and legend).
4. Make sure both your train_RNN.py and test_RNN.py files run by calling them from the command line without any extra inputs/arguments.

What to add in your report:

- Explanation of how you created your dataset.
- Any preprocessing steps you followed
- All design steps you went through in finding the best network in your report and how you chose your final design.
- Architecture of your final network, number of epochs, batch size (if needed), loss function, training algorithm, etc.
- Output of the training loop with comments on your output
- Output from testing, including the final plot and your comment on it
- What would happen if you used more days for features (feel free to actually try it – but do not upload the datasets).