COSI-149B-Project 2

Authors: MyongJoon Chang, Mark McAvoy, Eurey Noguchi

Project 2 was building a model which would take historical information about stock prices and using this data to predict the movement of stocks. Given the history up to time **t** our model will be predicting stock prices increase or decrease in the range of 2%, 3% and 5%.

To run the program run the Jupyter Notebook Main.ipynb and run the cells and type according to the instructions as printed as the Jupyter Notebook is run.

Our Notebook takes a single .csv file with test data information and use these data to predict the future movements.

Training:

During the training process of this project our group took the initial 3+ GB .csv file and broke up the files into multiple files according to its stock ID. We found several advantages in doing this, first off this saved us on running time during the training process, as it would reduce our overhead cost of having to read a larger file but would be constantly reading smaller files. During the process of breaking up our main test data into individual stock ID files, what we also did was we dropped what we considered unnecessary information, the columns which we thought would not be useful as input data. The columns which that we dropped where, Open, Close, High, Low, Volume, Dividend, Split, Adjusted Volume and ID. After trial and error we found that these information was not necessary as the adjusted values would be accounting for the dividends and splits. Additional to dropping the unnecessary columns we created an additional column called moving average which be getting an average of the 5 days moved into the future, our group found that this indicator was used a lot of in technical analysis for stock, therefore decided to add this column. Along with this since we scale the data from 0~1, this is a quite important stage as this is the part which lets us reuse the same model for making predictions of different stocks, if we did not scale as the absolute values would be all over the place our model would make very poor predictions.

Model:

Our group decided to use an LSTM model with four layers, with 50 units per layer with an output layer of 5 units, the 5 units predicted is represented by adjusted_open, adjusted_high, adjusted_low, adjusted_close, moving_average. For the input vector use 60 days of information to create one input vector, this allows model to use more information for training, and therefore give more accurate data along with this a more accurate prediction. The output layer would be one day's information for the 5 variables mentioned above.

To Replicate:

First input the Test Data .csv file name into the Python program clean_data.py and run the file. This will clean the data as mentioned above, after this run the Jupyter Notebook stock_prediction_all_file.ipynb, which will parse through all the .csv files created and train and adjust the weights of the model that we have created, saving after each stock ID is completed.

Additional code is provided at the bottom of the Notebook, which helped us visualize what was happening at the initial stages of this project. After the training is done, all you need to do is make sure that the model.h5 has been saved and can you this model to predict stock prices.

Contributions:

MyongJoon: model research, stock_prediction_all_files.ipynb, write up

Eurey: wrapper class (Main.ipynb), clean_data.py, misc coding

Mark: model research, factor_analysis (attempted way to predicting stock we did not use)