ML Final Project Readme

Brian Falkenstein

1 Dependencies

All of the code is written in Python for Python 3.7.7. More specifically, the code is compiled into Jupyter Notebooks, meaning **Anaconda** is required to run them. Overall, to run the training and testing code, the latest versions of the following repositories are required:

- Anaconda (comes with Numpy, sklearn, etc.)
- Pytorch

Additional plugins are required to generate the data. Both of these API's require registration keys (which are included in the code), and one of them, IEXCloud, requires payment. For this reason, I am including the data generated in the *data* folder (it is very small). You can still run the code to generate the data, its just that I am limited in the amount of queries I can make to IEXCloud, and am practically at the limit for the month. The latest versions of the following two libraries must be installed to generate the data (code in $fetch\_data.ipynb$):

- iexfinance (for Python)
- newsapi-python

Everything can be installed using pip in the Anaconda terminal.

2 Data

As mentioned earlier, the data is included in the submission in the *data* folder. $all\_data\_dvsizeX$ are the data files that are actually processed by the models, as they contain both the stock and the news data, with document feature size X. There is also a list of all the companies used in the study in this folder.

3 Running Instructions

The 4 different models tested, RNN with and without News and LSTM with and without news, are in separate files, with self explainable names. Each model file follows a similar format: hyperparameter and model definition at the top,

followed by separate training/testing procedures, followed by a cross validation procedure, and the final block of code is the combined training/testing procedure which re-generates the train-test split. Each module can be run with shift+tab. The modules which do the training/testing display metrics and graphs which show how the model performed.

There is also the $fetch\_data$ script. This script works from the list of companies, and fetches the news articles and stock data during a set time period (set at the top of the script) and saves it in a format which can be easily interpreted later (JSON). Note that you can run just the last 4 modules to re-open the already generated news and stock data, and re-generate the combined data with various document embedding settings (this is where embedding size is altered, or where a different embedding scheme could be implemented).