AI in Finance MMAI 823

Process Document

Contents

[**Plan** 2](#_Toc29109527)

[**Data Collection** 2](#_Toc29109528)

[**Data Exploration** 3](#_Toc29109529)

[**Process Steps** 3](#_Toc29109530)

[**Python Code** 4](#_Toc29109531)

[**Validation** 5](#_Toc29109532)

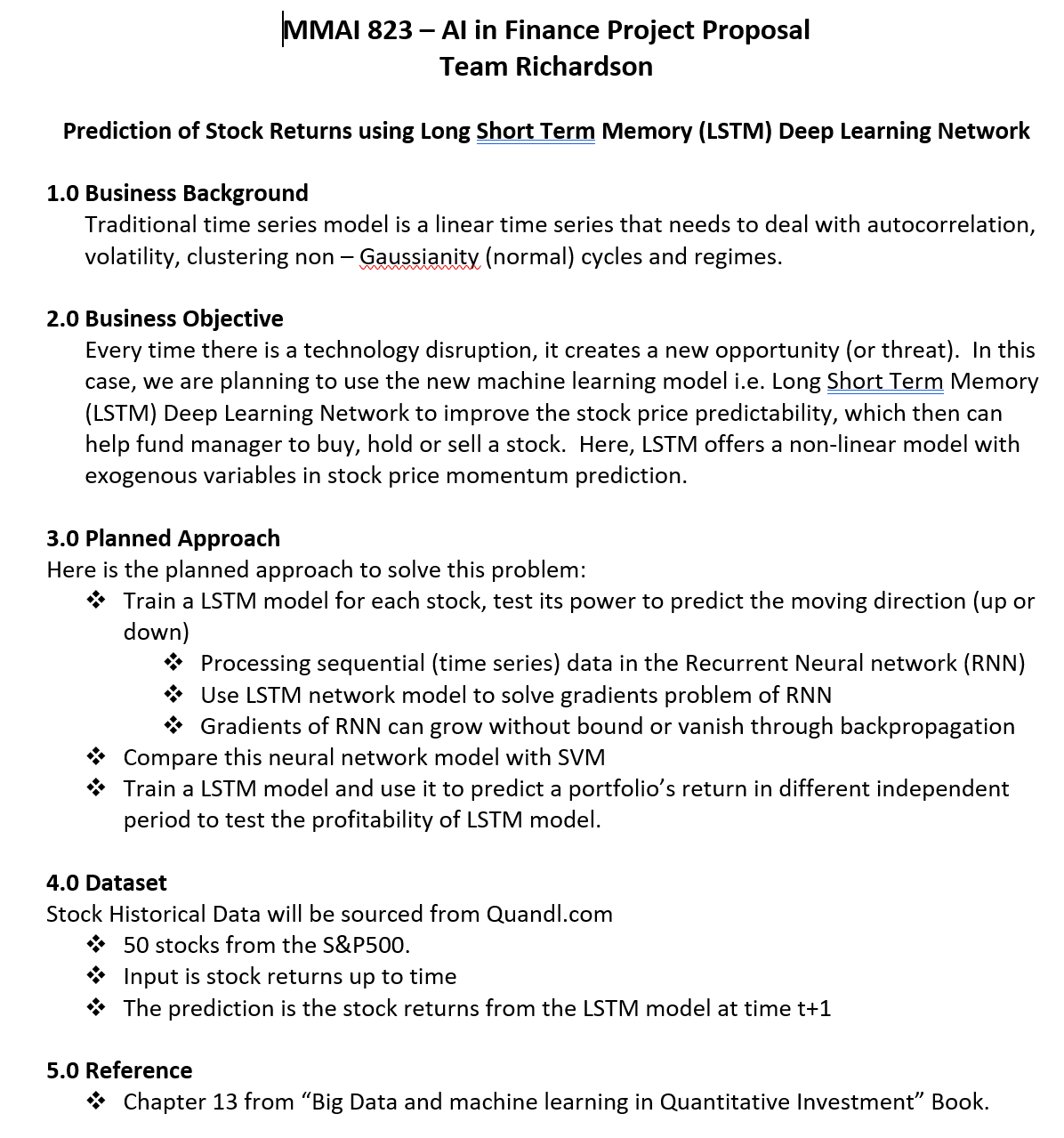
[**Articles** 6](#_Toc29109533)

[**Portfolio** 7](#_Toc29109534)

[**Next Steps** 8](#_Toc29109535)

[**Prerequisites for Back testing** 9](#_Toc29109536)

## **Plan**



## **Data Collection**

Haddy collected the data on different stocks. Here is the code that was used.

<https://github.com/Musyfy/AI_FIN_823/blob/master/Ingestion.ipynb>

<https://github.com/Musyfy/AI_FIN_823/blob/master/WIKI-FB.csv>

## **Data Exploration**

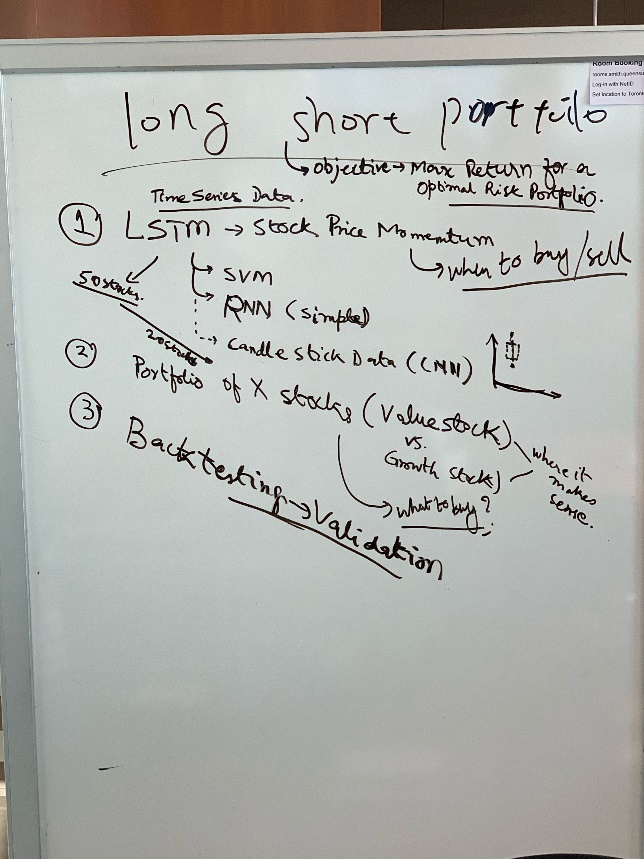
41 stocks

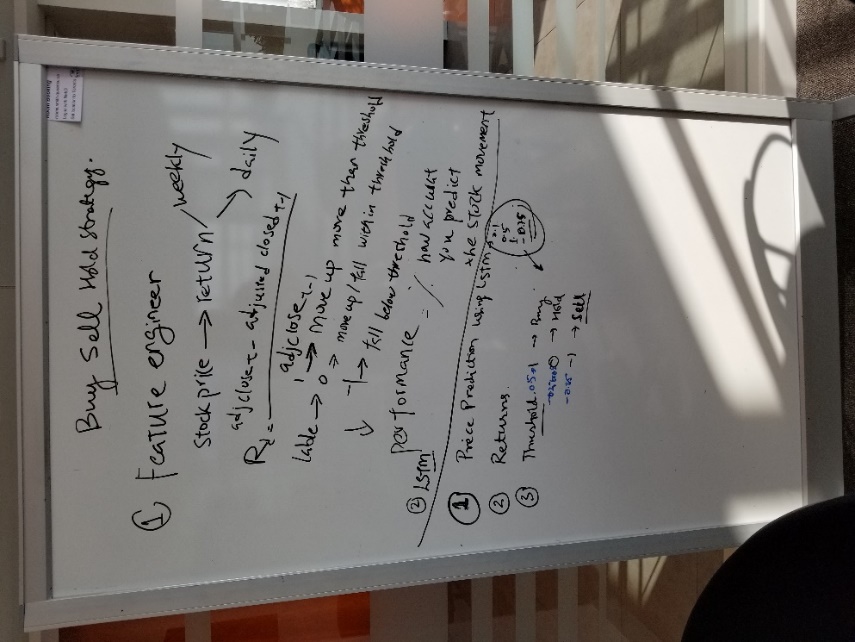
<https://github.com/Musyfy/AI_FIN_823/tree/master/MMAI%20823%20project/41%20stocks>

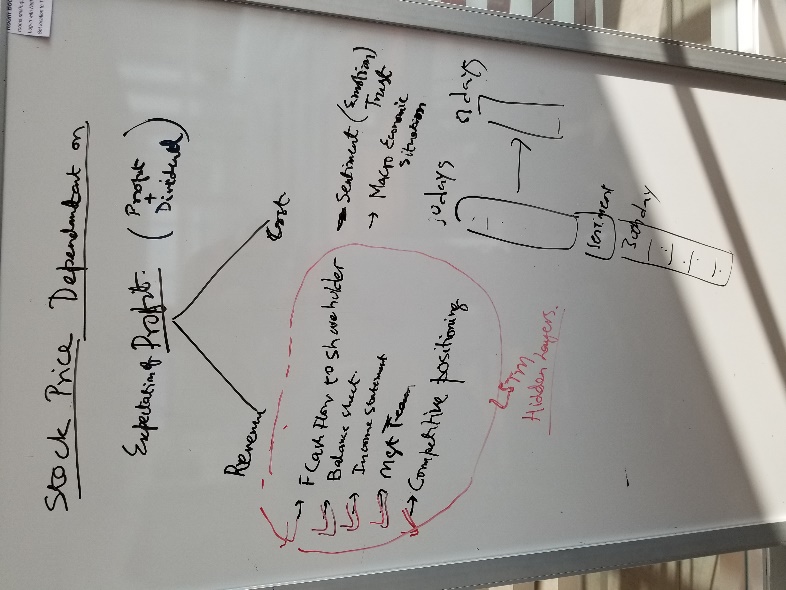
541 stocks

<https://github.com/Musyfy/AI_FIN_823/tree/master/MMAI%20823%20project/541%20stocks>

## **Process Steps**







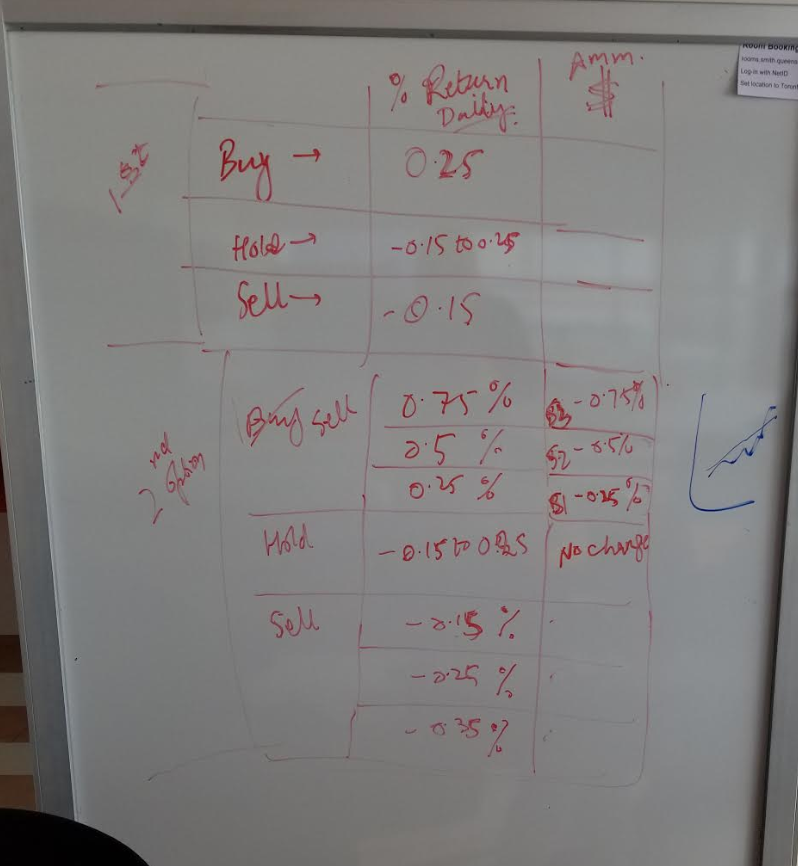
## **Python Code**

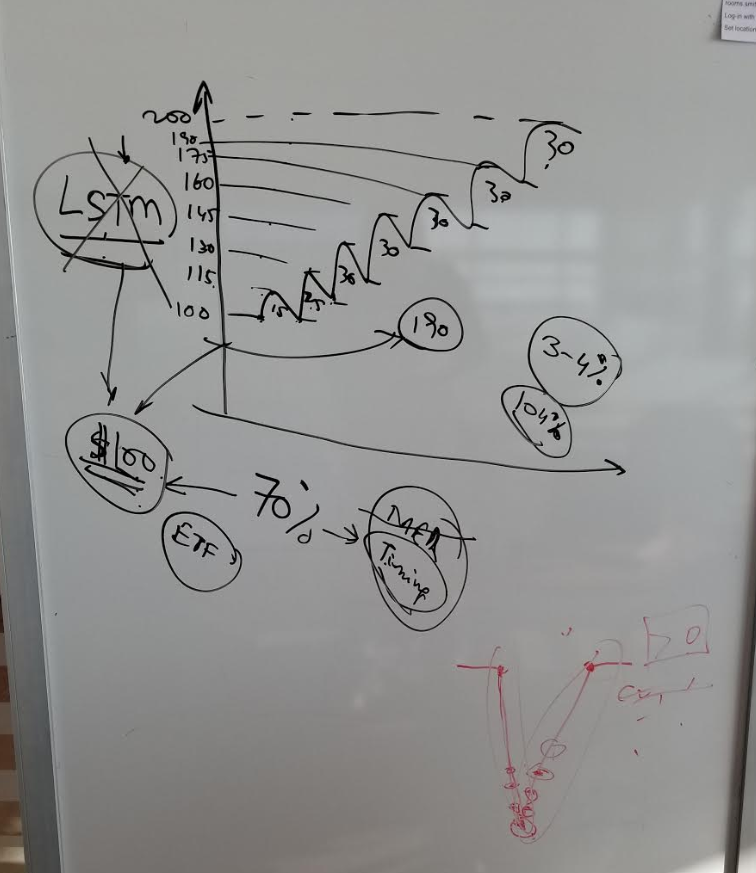
[LSTM Model.py](https://github.com/Musyfy/AI_FIN_823/blob/master/MMAI%20823%20project/LSTM%20Model.py) (Landis) (<https://github.com/Musyfy/AI_FIN_823/blob/master/MMAI%20823%20project/LSTM%20Model.py>)

## **Validation**

Three methods

1. Accuracy of our predictions – as a data scientist, we are interested in this. If we predicted the market to go up, and it went up, it’s True Positive.
2. Stock return – as an investor, we are interested in this. We can calculate the stock return over a period and compare it against the actuals. Here we have aother toggle in addition to the buy/ hold/ sell decision, that is how much to buy or sell. The LSTM model predicts the price movement i.e. when to buy or sell, but how much to sell is not part of the equation. Can we use the solver to determine the optimal amount to trade.





1. Stock Return - you can programmatically calculate the returns on the following site - [www.quantopian.com/](http://www.quantopian.com/)

## **Articles**

1. Deep Learning Frontier Method for Optimal Investments.pdf
2. ARIMA/ SARIMA vs. LSTM (RNN)

<https://towardsdatascience.com/arima-sarima-vs-lstm-with-ensemble-learning-insights-for-time-series-data-509a5d87f20a>

1. Using LSTMs For Stock Market Predictions (Tensorflow)

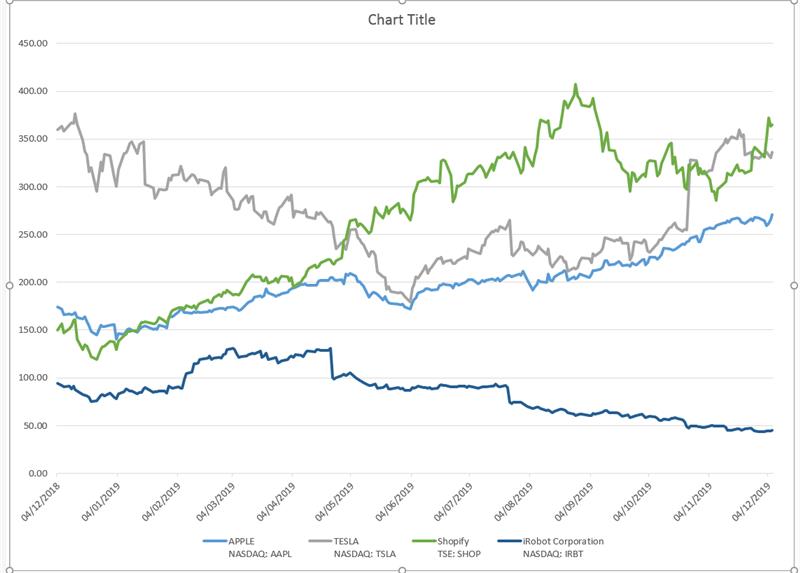
<https://towardsdatascience.com/using-lstms-for-stock-market-predictions-tensorflow-9e83999d4653>

1. Predicting stock prices using deep learning

<https://towardsdatascience.com/getting-rich-quick-with-machine-learning-and-stock-market-predictions-696802da94fe>

## **Portfolio**

[Apple Stocks Simualtion rev 3.xlsx](file:///D:\My%20Docs%20USB\Personal%20Files%20Jan%2018%202018\MMAI\19%20AI%20in%20Finance%20MMAI%20823\Assignments\Team%20Assignment\Team%20Project)



## **Next Steps**

|  |
| --- |
| What stock do we pick for our portfolio? |
| S&P.Less volatility stock, which sector we pick? |  |
| Sarb: manufacture 17 |  |
| Landis financial institution 17 |  |
| Haddy: technology 16 |  |
| 50 stocks |  |
|  |  |
| train these 50 models: |  |
| Do hyperparameter tunning on each stock |  |
| do it in the team; each team member pick 8,9 stocks |  |
| we aligned on the training data, testing data, and period. |  |
| training period: 2017-06-29 to 2019-06-29 |  |
| testing period:6 months 2019-06-30 to 2019-12-31 |  |
| (could change in the team meeting) |  |
| Aligned on the cost function: MSE |  |
|  |  |
| hyperparameters: |  |
| cost function |  |
| hidden layers |  |
| epochs |  |
| batch size |  |
| optimizer |  |
| Dropout ratio |  |
| learning rate |  |
| number of units of each layer |  |
| Window 15, 30, 45 |  |
| Predicting one day in advance is your target. |  |
|  |  |
| How should we decide the weights of each stock? |  |
| Equal-weighted?easy to implement |  |
| or select by apt model or camp model? |  |
|  |  |
| What the logic to do the trading to limit the risk and maximize the return? |  |

## **Prerequisites for Back testing**

Download Python 3.5 & Zipline

<https://github.com/Musyfy/AI_FIN_823/blob/master/MMAI%20823%20project/Backtesting/Zipline%20instruction.docx>

**Zipline instruction:**

Zipline only works on python 3.5

Since you need to install python 3.5 on your machine, and you will have multiple versions of python on your device.

**Virtual env:**

I suggest you create a virtual environment only for you python 3.5.

You need to add python 3.5 into your path and variable in your control panel.

Useful link to setup virtual env: <https://www.youtube.com/watch?v=APOPm01BVrk>

Then use this code in your command prompt:

 virtualenv -p path/to python35/python.exe your\_venvname

**Install Zipline:**

Then please follow the link to set up your zipline.

Useful link <https://pythonprogramming.net/zipline-local-install-python-programming-for-finance/>

Before you download all the required packages into your local computer, don’t forget to activate your virtual environment.

After you install all required packages in your local computer, use the following code in your command prompt

pip install Your-package’s name.whl

**Final step:**

Now you have zipline in your virtual environment. Since the virtual environment is an independent block in your computer, it wouldn’t be able to call the other packages that you previously installed on python 3.6.

You need to install your pandas, numpy, jupyter notebook, matplot, and quandl packages again in your virtual environment.

Then you can run your jupyter notebook and testing the scripts I sent.

## **Check/ Clarification**

Comparison Models

* LSTM
* Moving Average
* Exponential Moving Average
* SVM