| Nguyen, Duc Trung | 202383238
| Thomson, Alasdair | 202383138

1. Background to the problem

In this assignment, we will try to develop a trading bot focus on short-term investment of blue chip stocks. For this, we wil perform 2 tasks:

(1) Applying **Neural Network**, specifically the Long Short-Term Memory (LSTM) Neural network, using previous days closing price and other technical indicators as input, to make next day price prediction

(2) Applying **Genetic Algorithm** to develop an efficient trading strategy that can utlize the price prediction, maximize profit while also maintain certain level of flexibility.

For task 1, we will be using various technical indicators including:

1. **Simple Moving Average (SMA)**: SMA calculates the average price of a security over a specific period, helping to smooth out price fluctuations and identify trends.
2. **Exponential Moving Average (EMA)**: EMA is a type of moving average that gives more weight to recent prices, making it more responsive to recent price changes and useful for trend identification.
3. **Relative Strength Index (RSI)**: RSI is a momentum oscillator ranging from 0 to 100, used to identify overbought or oversold conditions in a security based on recent price movements.
4. **Moving Average Convergence Divergence (MACD)**: MACD is a trend-following momentum indicator that compares two moving averages of a security's price to identify bullish and bearish signals and potential trend reversals.

We will be using data of a short time period (previous 1 to 5 days), but by incorporating these technical indicators, our predictions would based on infomation of a much bigger time window while still maintain relatively fast.

We will be building on other research utilizing LSTM and using many of the techniques identified in this paper (https://www.analyticsvidhya.com/blog/2021/12/stock-price-prediction-using-lstm/).

1.1 A note on the use of python

Please note that this assignment is delivered in Python with the permission of the professor. This is due to our use of a neural network in creating predictions which we found was easier to perform in Python.

Show code

```
python version:  3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
```

2. Overview of chosen data

For our analysis, we will be working with the top 20 stocks by capitalization from the S&P 500. The main reason for this choice is because:

(1) S&P 500 stocks are the most liquid stocks in the markets, which would allow use to act on our prediction with minimal delay

(2) S&P 500 companies are subject to high regulatory compliance with high level of capitalization, which make these stocks less subceptible for malpractices and market manimpulation comparing to others

We will be using the data from 9/2015 to 12/2019, in which the 4 months period in 2015 will be use as buffer for calculating the technical indicators, data of the period 2016 to 2018 are use for training and 2019 will be use for testing purpose.

2.1. Importing the library

```
pip install yfinance
```

```
pip install deap
```

Show hidden output

Show code

```
yfinance version: 0.2.37
Pandas version: 2.0.3
NumPy version: 1.25.2
Matplotlib version: 3.7.1
Scikit-learn version: 1.2.2
TensorFlow version: 2.15.0
```

2.2. Importing stock data

Show code

Show hidden output

Show code

	Symbol	Name	Sector
0	AAPL	Apple	Information Technology

1	ABBV	AbbVie	Health Care
9	AMZN	Amazon	Consumer Discretionary
20	CAT	Caterpillar	Industrials
33	DIS	Disney	Communication Services
44	GOOGL	Alphabet (Class A)	Communication Services
46	HD	Home Depot	Consumer Discretionary
49	INTC	Intel	Information Technology
51	JNJ	Johnson & Johnson	Health Care
52	JPM	JPMorgan Chase	Financials
54	KO	Coca-Cola	Consumer Staples
59	MA	Mastercard	Information Technology
67	MRK	Merck	Health Care
69	MSFT	Microsoft	Information Technology
76	PFE	Pfizer	Health Care
77	PG	Procter & Gamble	Consumer Staples
86	T	AT&T	Communication Services
92	UNH	UnitedHealth Group	Health Care
96	V	Visa	Information Technology
97	VZ	Verizon	Communication Services

As our trading bot will only consider buy and sell, and not including complex activities like short-selling, so we choose stocks from various industries to maximize investment opportunities even under market down trend.

```
df = SP_stock_data.copy()
```

Show code

The average closing price of our basket of 20 stocks are present in blue and use the y-axis on the left, while the S&P500 index are presented in red and use the y-axis on the right. We can see that the 2 figures are heavily correlated, since the 20 stocks make up a large portion in term of capitalization of the S&P500 and cover different sectors. We will be using the period from 2016 to end of 2018 for training, and 2019 for testing. While there is a major market downtrend around 12/2018, the overal market movement of these 2 time period are both upward with similar seasonality.

Show code

```
Percentage change of S&P500 from start of 2016 to end of 2018: 22.65%
Percentage change of S&P500 from start to end of 2019: 27.01%
Percentage change of average prices from start of 2016 to end of 2018: 49.86%
Percentage change of average prices from start to end of 2019: 66.63%
```

Based on the above figure, we can see that the training period profit was heavily affect by the 12/2018 downtrend, and the basket of 20 stocks we picked has a relatively better performance compare to the S&P.

Show code

2.3. Perform feature engineer

We calculate various financial technical indicators and compare the result to see if using additional data help with the prediction process. Since we will be using LSTM, a form of recurrent neural network, and as neural network typically can perform feature engineering by itself, so this step can typically be skip. The reason we are doing this is to incorporate information of stock's momentum in a longer period (between 14 to 50 days) than the time window we will be using (1 and 5 days) into the model.

Show code

Show code

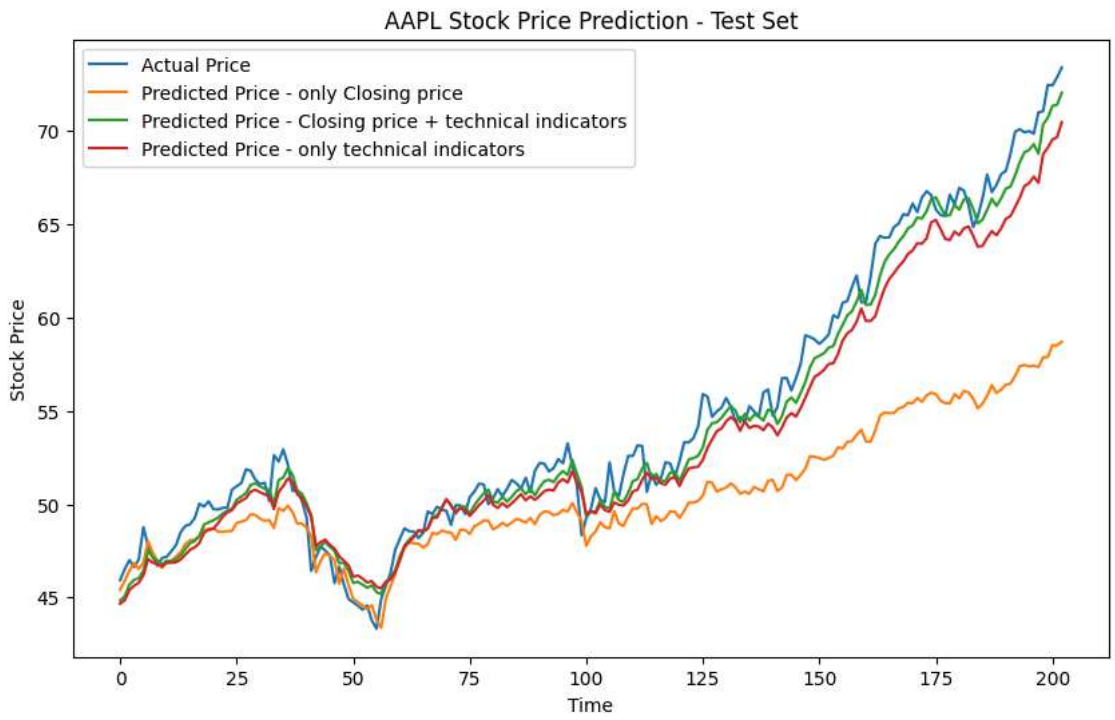
2.4. Predict stock price using LSTM

2.4.1. Predict using time_step = 1

Show code

Show code

```
7/7 [==============================] - 0s 4ms/step
7/7 [==============================] - 0s 5ms/step
7/7 [==============================] - 0s 4ms/step
Test Mean Squared Error for Close feature: 34.47169705504316
Test Mean Squared Error for All features: 1.212663186801293
Test Mean Squared Error for Technical features: 2.904653643020139
```



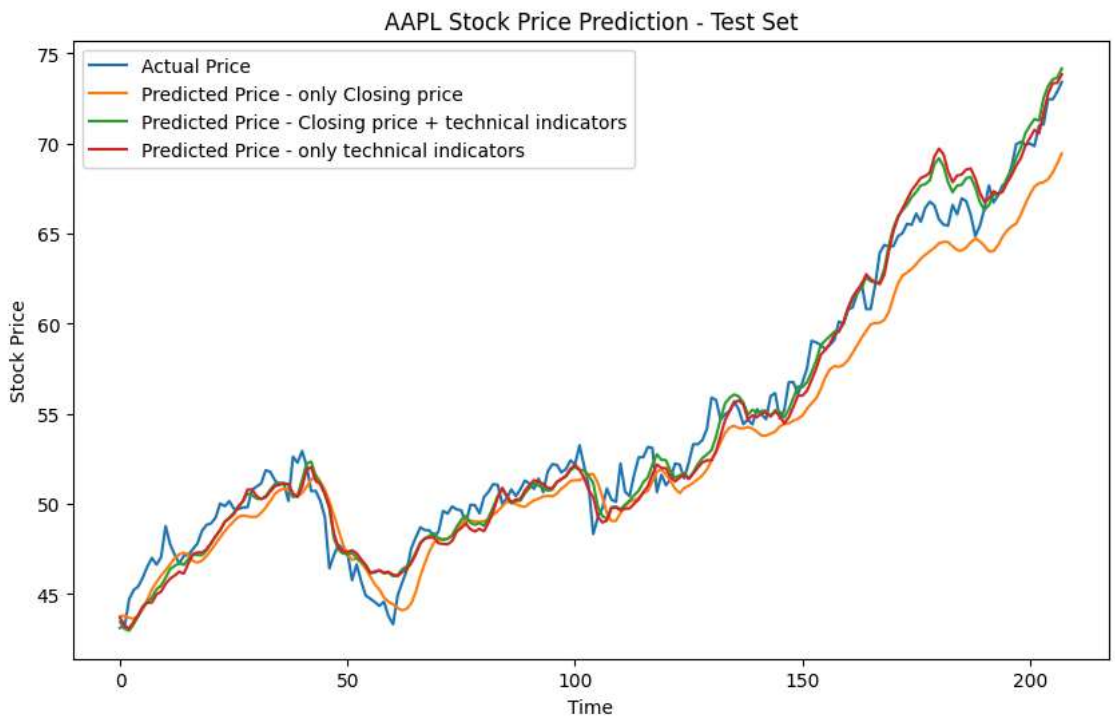
2.4.2. Predict using time_step = 5

Show code

```
7/7 [==============================] - 0s 8ms/step
7/7 [==============================] - 0s 8ms/step
7/7 [==============================] - 0s 8ms/step
```

Show code

```
Test Mean Squared Error for Close feature: 3.6492508098265786
Test Mean Squared Error for All features: 1.468306868091811
Test Mean Squared Error for Technical features: 1.802232411033961
```



We performed prediction 2 time window of 1 and 5 trading days (equivalent to 1 week) with 3 combination of data which are:

- Close: just using previous day(s) closing price of the stock
- All: using previous day(s) closing price along with 10 technical indicators
- Technical: just using the 10 technical indcators

Overall, We observe that adding the technical indicators into the model does improve the prediction. And while using a time window of 5 days slightly reduce the All prediction, as this seem to improve the Close and Technical prediction, we will be training the prediction model using 5 days of All data to predict the next day Closing price.

2.5. Applying the model to all of the stocks

Show code

Show code

We will loop through the 20 stocks, train 20 models (1 for each stocks), then using the resulting models to perform 2 predictions: (1) for the training period and (2) for the testing period

Show code

```
Processing AAPL...
24/24 [==============================] - 1s 8ms/step
AAPL - Training Period MSE: 1.1511836915712435
8/8 [==============================] - 0s 9ms/step
AAPL Test MSE: 2.9938854703511435
Processing MSFT...
24/24 [==============================] - 1s 9ms/step
MSFT - Training Period MSE: 1.7759610579434473
8/8 [==============================] - 0s 7ms/step
MSFT Test MSE: 18.24175733938072
Processing AMZN...
24/24 [==============================] - 1s 7ms/step
AMZN - Training Period MSE: 3.226963040242966
8/8 [==============================] - 0s 6ms/step
AMZN Test MSE: 4.203007183394365
Processing ABBV...
24/24 [==============================] - 1s 7ms/step
ABBV - Training Period MSE: 3.933995855152044
8/8 [==============================] - 0s 7ms/step
ABBV Test MSE: 3.8564977898992123
Processing GOOGL...
24/24 [==============================] - 1s 8ms/step
GOOGL - Training Period MSE: 0.7622460583656211
8/8 [==============================] - 0s 8ms/step
GOOGL Test MSE: 2.1000438270313744
Processing KO...
24/24 [==============================] - 1s 7ms/step
KO - Training Period MSE: 0.5478419799247835
8/8 [==============================] - 0s 7ms/step
KO Test MSE: 11.761401694987988
Processing CAT...
24/24 [==============================] - 0s 6ms/step
CAT - Training Period MSE: 9.052950049072251
8/8 [==============================] - 0s 6ms/step
CAT Test MSE: 16.0670541816148
Processing JNJ...
24/24 [==============================] - 0s 7ms/step
JNJ - Training Period MSE: 2.6094382292035347
8/8 [==============================] - 0s 8ms/step
JNJ Test MSE: 3.2443722499851266
Processing JPM...
24/24 [==============================] - 1s 6ms/step
JPM - Training Period MSE: 2.6516335949880694
8/8 [==============================] - 0s 8ms/step
JPM Test MSE: 5.817569402243155
Processing V...
24/24 [==============================] - 0s 7ms/step
V - Training Period MSE: 3.5757033250979147
8/8 [==============================] - 0s 6ms/step
V Test MSE: 16.1072294087895
Processing PG...
24/24 [==============================] - 0s 6ms/step
PG - Training Period MSE: 0.9206048982972279
8/8 [==============================] - 0s 7ms/step
PG Test MSE: 24.81401505015361
Processing UNH...
24/24 [==============================] - 1s 11ms/step
UNH - Training Period MSE: 12.90242728123938
```

2.6. Evaluate the prediction

Show code

```
   Stocks        MSE
16      T   0.202206
15     VZ   0.449939
5      KO   0.671687
4   GOOGL   0.743993
10     PG   0.961874
0    AAPL   1.071749
19    PFE   1.490650
1    MSFT   1.786262
18    MRK   1.825797
12   INTC   2.006049
17    DIS   2.010660
8     JPM   2.151407
7     JNJ   2.524676
9       V   3.082261
2    AMZN   3.247305
3    ABBV   4.018234
6     CAT   8.958984
13     HD   9.572498
11    UNH  12.104260
14     MA  14.108120
Average MSE across all tickers: 3.64943058512639
Median MSE across all tickers: 2.008354703836376
```

Show code

```
   Stocks        MSE
16      T   0.347508
15     VZ   1.261946
0    AAPL   2.535253
7     JNJ   3.480332
19    PFE   3.531944
4   GOOGL   3.607010
3    ABBV   3.957313
8     JPM   4.583970
12   INTC   6.011559
2    AMZN   8.660538
5      KO  11.043133
6     CAT  17.360109
10     PG  20.009045
9       V  20.903300
1    MSFT  24.482488
18    MRK  43.265220
11    UNH  55.595540
17    DIS  63.319656
14     MA  79.358931
13     HD  95.828879
Average MSE across all tickers: 23.45718364760329
Median MSE across all tickers: 9.8518356511353
```

3. Developing trading strategy

Agent Trading an individual stock

Set up data

Evaluation setup

Show code

Trading Rules

The agent buys when there is a buy indicator Pre_Binary_Indicator, and sells when it's negative

Single trader

Show code

```
Bought 277.0 shares at: $35.99 on 2019-01-03 00:00:00, Remaining cash: $29.39
Sold 0 shares at: $40.81 on 2019-01-30 00:00:00, Cash after sale: $11334.45
Bought 262.0 shares at: $43.10 on 2019-02-07 00:00:00, Remaining cash: $42.25
Sold 0 shares at: $42.53 on 2019-02-12 00:00:00, Cash after sale: $11183.80
Bought 254.0 shares at: $43.99 on 2019-03-05 00:00:00, Remaining cash: $11.61
Sold 0 shares at: $43.87 on 2019-03-11 00:00:00, Cash after sale: $11155.22
Bought 232.0 shares at: $47.88 on 2019-03-25 00:00:00, Remaining cash: $47.64
Sold 0 shares at: $47.19 on 2019-03-27 00:00:00, Cash after sale: $10995.14
Bought 232.0 shares at: $47.24 on 2019-03-28 00:00:00, Remaining cash: $36.04
Sold 0 shares at: $47.91 on 2019-04-01 00:00:00, Cash after sale: $11151.16
Bought 222.0 shares at: $50.21 on 2019-04-11 00:00:00, Remaining cash: $3.99
Sold 0 shares at: $51.11 on 2019-04-23 00:00:00, Cash after sale: $11349.85
Bought 218.0 shares at: $51.84 on 2019-04-24 00:00:00, Remaining cash: $48.73
Sold 0 shares at: $52.47 on 2019-05-01 00:00:00, Cash after sale: $11487.19
Bought 224.0 shares at: $51.07 on 2019-05-06 00:00:00, Remaining cash: $46.95
Sold 0 shares at: $46.57 on 2019-05-15 00:00:00, Cash after sale: $10478.07
Bought 228.0 shares at: $45.88 on 2019-05-20 00:00:00, Remaining cash: $17.43
Sold 0 shares at: $46.63 on 2019-06-07 00:00:00, Cash after sale: $10648.50
Bought 219.0 shares at: $48.49 on 2019-06-12 00:00:00, Remaining cash: $29.74
Sold 0 shares at: $48.22 on 2019-06-17 00:00:00, Cash after sale: $10591.02
Bought 213.0 shares at: $49.61 on 2019-06-25 00:00:00, Remaining cash: $24.62
Sold 0 shares at: $49.44 on 2019-06-26 00:00:00, Cash after sale: $10555.87
Bought 210.0 shares at: $50.20 on 2019-07-08 00:00:00, Remaining cash: $13.35
Sold 0 shares at: $50.46 on 2019-07-10 00:00:00, Cash after sale: $10610.47
Bought 207.0 shares at: $51.01 on 2019-07-17 00:00:00, Remaining cash: $50.88
Sold 0 shares at: $51.00 on 2019-07-18 00:00:00, Cash after sale: $10607.88
Bought 206.0 shares at: $51.45 on 2019-07-19 00:00:00, Remaining cash: $9.70
Sold 0 shares at: $50.91 on 2019-07-22 00:00:00, Cash after sale: $10497.67
Bought 201.0 shares at: $52.19 on 2019-07-30 00:00:00, Remaining cash: $7.48
Sold 0 shares at: $54.10 on 2019-07-31 00:00:00, Cash after sale: $10882.59
Bought 203.0 shares at: $53.47 on 2019-08-01 00:00:00, Remaining cash: $27.17
Sold 0 shares at: $48.85 on 2019-08-07 00:00:00, Cash after sale: $9944.22
Bought 189.0 shares at: $52.36 on 2019-08-23 00:00:00, Remaining cash: $48.66
Sold 0 shares at: $52.12 on 2019-08-29 00:00:00, Cash after sale: $9900.28
Bought 191.0 shares at: $51.61 on 2019-09-03 00:00:00, Remaining cash: $43.25
Sold 0 shares at: $52.10 on 2019-09-04 00:00:00, Cash after sale: $9993.87
Bought 181.0 shares at: $55.00 on 2019-09-13 00:00:00, Remaining cash: $38.87
Sold 0 shares at: $55.22 on 2019-09-30 00:00:00, Cash after sale: $10034.60
Bought 179.0 shares at: $55.76 on 2019-10-02 00:00:00, Remaining cash: $52.66
Sold 0 shares at: $54.61 on 2019-10-03 00:00:00, Cash after sale: $9827.40
Bought 174.0 shares at: $56.46 on 2019-10-08 00:00:00, Remaining cash: $4.23
Sold 0 shares at: $56.98 on 2019-10-10 00:00:00, Cash after sale: $9919.19
Bought 167.0 shares at: $59.10 on 2019-10-15 00:00:00, Remaining cash: $49.90
Sold 0 shares at: $62.38 on 2019-11-01 00:00:00, Cash after sale: $10468.20
Bought 162.0 shares at: $64.26 on 2019-11-05 00:00:00, Remaining cash: $57.68
Sold 0 shares at: $66.87 on 2019-12-06 00:00:00, Cash after sale: $10890.62
Bought 161.0 shares at: $67.50 on 2019-12-09 00:00:00, Remaining cash: $23.12
Final sell 161.0 shares at: $72.48, Final cash: $11692.80
Total profit from trading AAPL: $1692.80
```

Agent trading across all stocks

Trading Rules

Now the agent looks at every stock to find which is going up the most while also selling upon a negative indicator. A holding period was also introduced to account for the uncertainty and rapidly changing buy and sell signals

Multistock trader

Show code

```
Bought 277 shares of AAPL at $35.99 on 2019-01-03 00:00:00, remaining cash: $29.39
Bought 1 shares of T at $23.32 on 2019-01-09 00:00:00, remaining cash: $6.06
Sold 277 shares of AAPL at $40.81 on 2019-01-30 00:00:00, cash after sale: $11311.12
Bought 145 shares of ABBV at $77.69 on 2019-01-30 00:00:00, remaining cash: $46.07
Bought 1 shares of AAPL at $43.10 on 2019-02-07 00:00:00, remaining cash: $2.97
Sold 1 shares of AAPL at $42.53 on 2019-02-12 00:00:00, cash after sale: $45.50
Sold 1 shares of T at $22.68 on 2019-02-15 00:00:00, cash after sale: $68.18
Bought 1 shares of GOOGL at $56.97 on 2019-02-15 00:00:00, remaining cash: $11.22
Sold 1 shares of GOOGL at $55.72 on 2019-02-26 00:00:00, cash after sale: $66.93
Bought 1 shares of KO at $45.10 on 2019-02-26 00:00:00, remaining cash: $21.83
Sold 1 shares of KO at $45.53 on 2019-03-04 00:00:00, cash after sale: $67.36
Bought 1 shares of VZ at $57.15 on 2019-03-04 00:00:00, remaining cash: $10.21
Sold 1 shares of VZ at $56.76 on 2019-03-11 00:00:00, cash after sale: $66.97
Bought 2 shares of T at $22.64 on 2019-03-11 00:00:00, remaining cash: $21.69
Sold 145 shares of ABBV at $80.06 on 2019-03-15 00:00:00, cash after sale: $11630.39
Bought 194 shares of GOOGL at $59.90 on 2019-03-15 00:00:00, remaining cash: $9.79
Sold 194 shares of GOOGL at $60.07 on 2019-03-20 00:00:00, cash after sale: $11663.37
Bought 144 shares of ABBV at $80.72 on 2019-03-20 00:00:00, remaining cash: $39.69
Sold 144 shares of ABBV at $79.85 on 2019-03-26 00:00:00, cash after sale: $11538.09
Bought 240 shares of AAPL at $47.92 on 2019-03-26 00:00:00, remaining cash: $38.49
Sold 240 shares of AAPL at $47.91 on 2019-04-01 00:00:00, cash after sale: $11536.89
Bought 142 shares of ABBV at $80.98 on 2019-04-01 00:00:00, remaining cash: $37.73
Sold 142 shares of ABBV at $83.70 on 2019-04-08 00:00:00, cash after sale: $11923.12
Bought 196 shares of GOOGL at $60.56 on 2019-04-08 00:00:00, remaining cash: $54.25
Bought 1 shares of KO at $46.54 on 2019-04-08 00:00:00, remaining cash: $7.71
Sold 1 shares of KO at $46.88 on 2019-04-15 00:00:00, cash after sale: $54.59
Bought 1 shares of AAPL at $49.65 on 2019-04-15 00:00:00, remaining cash: $4.94
Sold 1 shares of AAPL at $51.11 on 2019-04-23 00:00:00, cash after sale: $56.05
Bought 1 shares of AAPL at $51.84 on 2019-04-24 00:00:00, remaining cash: $4.21
Sold 1 shares of AAPL at $52.47 on 2019-05-01 00:00:00, cash after sale: $56.68
Bought 1 shares of INTC at $51.10 on 2019-05-01 00:00:00, remaining cash: $5.58
Sold 196 shares of GOOGL at $58.63 on 2019-05-02 00:00:00, cash after sale: $11497.06
Bought 120 shares of AMZN at $95.67 on 2019-05-02 00:00:00, remaining cash: $17.08
Sold 120 shares of AMZN at $95.94 on 2019-05-08 00:00:00, cash after sale: $11530.30
Bought 228 shares of AAPL at $50.47 on 2019-05-08 00:00:00, remaining cash: $22.00
Sold 2 shares of T at $22.85 on 2019-05-10 00:00:00, cash after sale: $67.69
Bought 1 shares of GOOGL at $58.44 on 2019-05-10 00:00:00, remaining cash: $9.25
Sold 228 shares of AAPL at $46.57 on 2019-05-15 00:00:00, cash after sale: $10626.64
Bought 84 shares of CAT at $125.91 on 2019-05-15 00:00:00, remaining cash: $50.20
Sold 1 shares of GOOGL at $58.59 on 2019-05-16 00:00:00, cash after sale: $108.79
Bought 4 shares of T at $23.68 on 2019-05-16 00:00:00, remaining cash: $14.08
Sold 4 shares of T at $24.02 on 2019-06-06 00:00:00, cash after sale: $110.15
Bought 2 shares of AAPL at $45.77 on 2019-06-06 00:00:00, remaining cash: $18.61
Sold 2 shares of AAPL at $48.72 on 2019-06-11 00:00:00, cash after sale: $116.04
Bought 1 shares of ABBV at $77.49 on 2019-06-11 00:00:00, remaining cash: $38.55
Bought 1 shares of T at $24.21 on 2019-06-11 00:00:00, remaining cash: $14.34
Sold 84 shares of CAT at $128.40 on 2019-06-18 00:00:00, cash after sale: $10799.94
Bought 80 shares of MSFT at $134.19 on 2019-06-18 00:00:00, remaining cash: $64.74
Bought 1 shares of GOOGL at $55.36 on 2019-06-19 00:00:00, remaining cash: $9.38
Sold 80 shares of MSFT at $137.00 on 2019-06-24 00:00:00, cash after sale: $10969.38
Bought 114 shares of AMZN at $95.63 on 2019-06-24 00:00:00, remaining cash: $67.21
Bought 1 shares of AAPL at $49.61 on 2019-06-25 00:00:00, remaining cash: $17.61
Sold 1 shares of ABBV at $68.16 on 2019-06-27 00:00:00, cash after sale: $85.77
Sold 1 shares of T at $24.95 on 2019-06-28 00:00:00, cash after sale: $110.72
Sold 1 shares of AAPL at $50.79 on 2019-07-01 00:00:00, cash after sale: $161.51
Bought 1 shares of CAT at $139.30 on 2019-07-01 00:00:00, remaining cash: $22.21
Sold 1 shares of GOOGL at $55.24 on 2019-07-02 00:00:00, cash after sale: $77.46
Bought 3 shares of T at $25.28 on 2019-07-02 00:00:00, remaining cash: $1.62
```

Implementing a Genetic Program for trading

GP

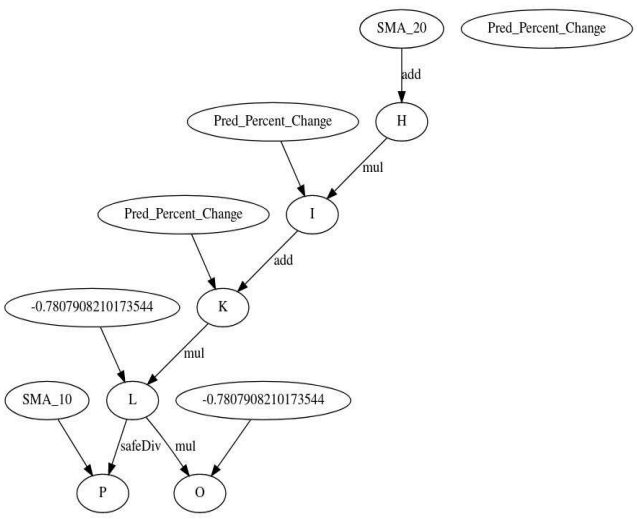
Show code

GP Execution

Show code

Please note that the output has been copied from another jupyter system due to time constraints

```
gen     nevals  avg     std     min             max
0       140     2970.93 3068.19 -6465.02        18680.7
1       87      4197.55 3683.28 -6465.02        18680.7
2       77      5208.69 3112.19 -4078.73        18680.7
3       80      5981.36 6298.49 -6465.02        44062.8
4       88      5906.84 7946.04 -6465.02        44062.8
5       95      6790.89 9410.68 -6465.02        44062.8
6       92      7688.78 9336.13 -6465.02        44062.8
7       81      9707.41 11229   -6465.02        44062.8
8       92      11338.6 12354   -6465.02        44062.8
9       101     10303   13983.2 -6465.02        44277.7
10      85      15180.8 16719   -6465.02        44277.7
11      81      19337.2 18322.6 -6465.02        44277.7
12      77      22045.2 19772.3 -6465.02        44277.7
13      84      24702.8 20773.3 -6465.02        44279.3
14      82      24730.1 20527.5 -6469.01        44277.7
15      81      26914.6 20543.6 -6469.01        44277.7
Final Testing Portfolio Value: 1138.1877517700195
```

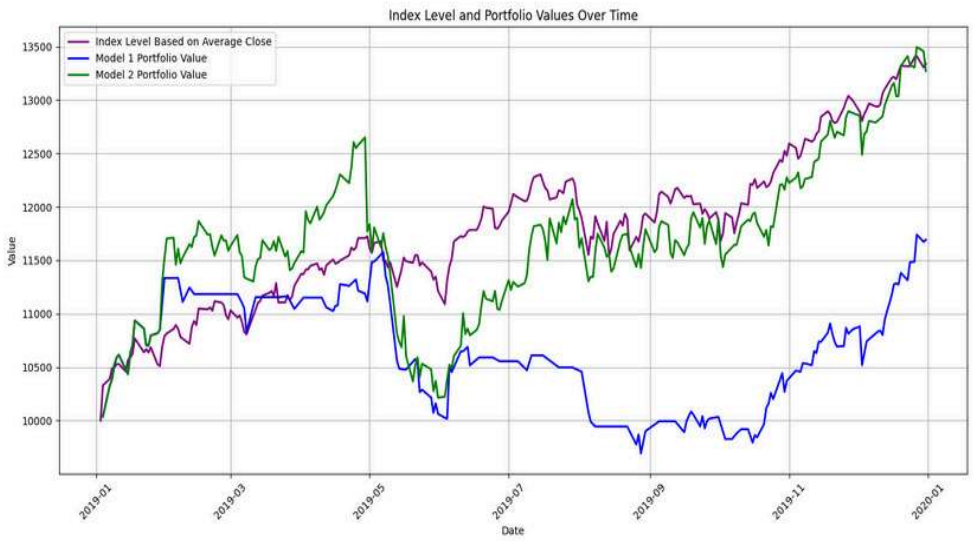


Trading Strategy comments and findings

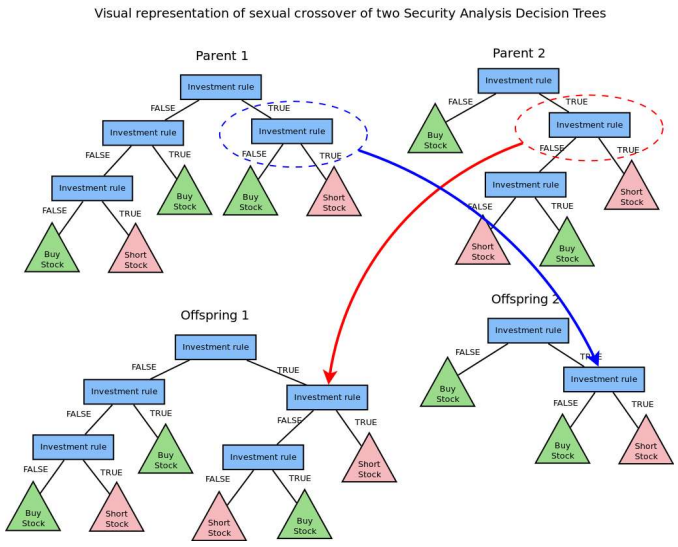
There were several approaches to trading. Our prediction based method unfortunately underperformed for the single stock, due to the noise in our predictions. They especially underperformed the market index which rose some 30% during the period. Our multi-stock model performed better, close to the market. Since this model heavily relied on predictions, this validates that our predictive neural network performed relatively well.

Our GP which underperformed, was not included, but nontheless beat inflation while underperforming relative to the other methods.

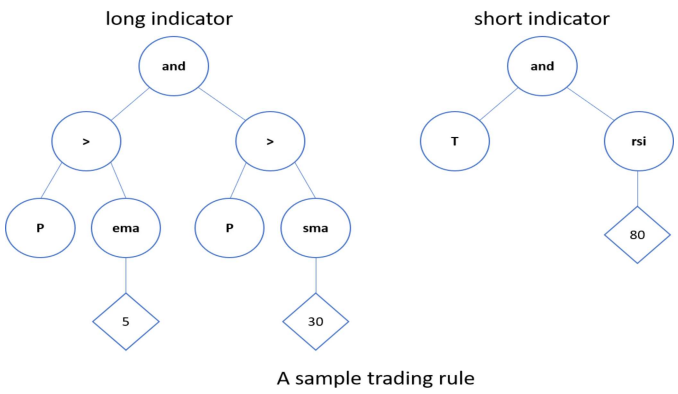
Show code



Our approach above is just one way to apply evolutionary programming into developing trading strategies. While generating buy and sell signal of a single stock is relatively simple and can potentially generate steady profits, in an open market we are require to make decision to choose our holding within various stocks, and this would require a complicate pre-determined trading rule to make trading bot applicable. For this problem, we can apply genetic programming to create trading strategies that can be present in the form of logic tree as presented below

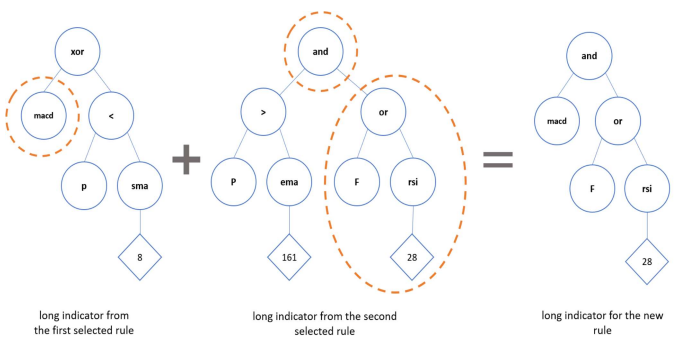


Based on the principle of genetic programming, the process can follow these steps. First, we present the strategies in the form of logic trees

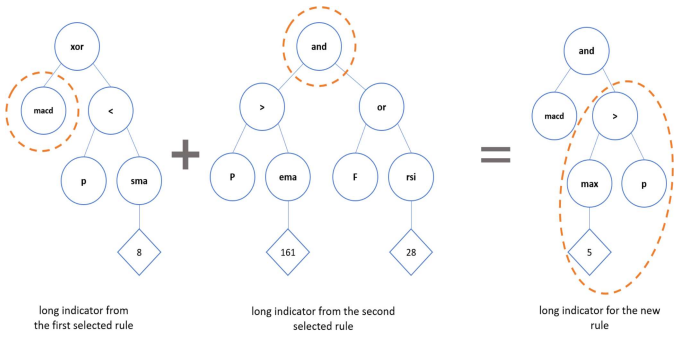


A sample trading rule

We can apply crossover between different rules



And mutate the resulting strategy



The resulting trading strategies can be more details.

```
> rule_printer(rule,full=TRUE)
buy rule:
(1) or.fn
   (2) larger.fn
      (3) SMA(Low 158)
      (3) SMA(High 124)
   (2) and.fn
      (3) larger.fn
         (4) Low
         (4) SMA(Low 138)
      (3) or.fn
         (4) KST.fn
         (4) larger.fn
            (5) runMin(High 129)
            (5) SMA(Low 192)
sell rule:
(1) xor
   (2) smaller.fn
      (3) EMA(High 25)
      (3) SMA(Close 126)
   (2) MACD.fn(12 26 9)
```

References

Turing Finance. "Using genetic programming to evolve security analysis decision trees." Turing Finance, https://www.turingfinance.com/using-genetic-programming-to-evolve-security-analysis-decision-trees/.

Zelin Chen. Genetic Programming trading rules searching. 2018. GitHub, https://github.com/username/repository.