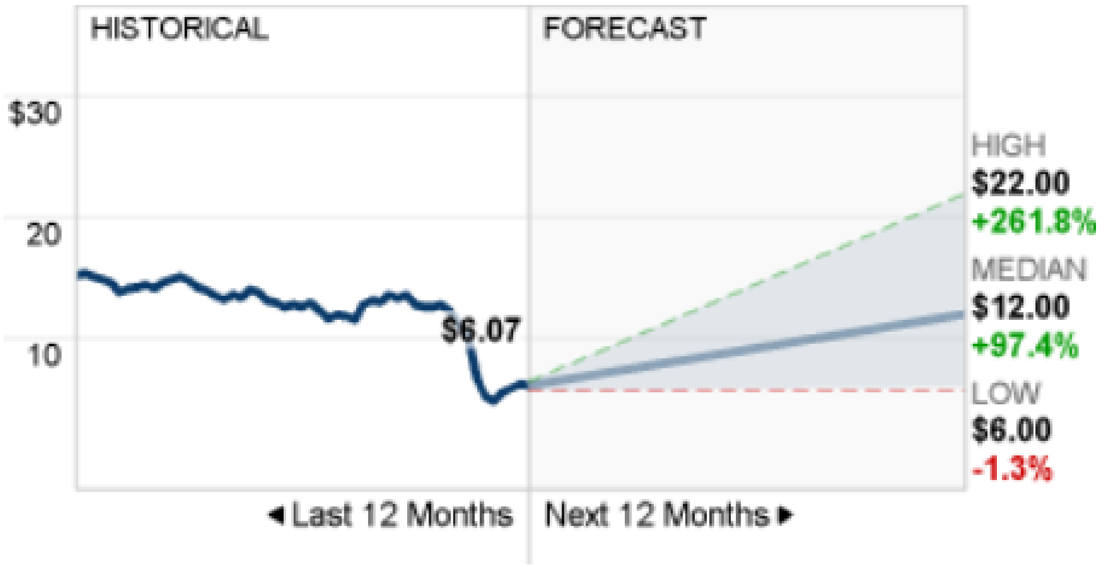
Capstone Project: Analysis of Stock price and forecasting using LSTM

Rezoan Ahmed Shuvro

Objective

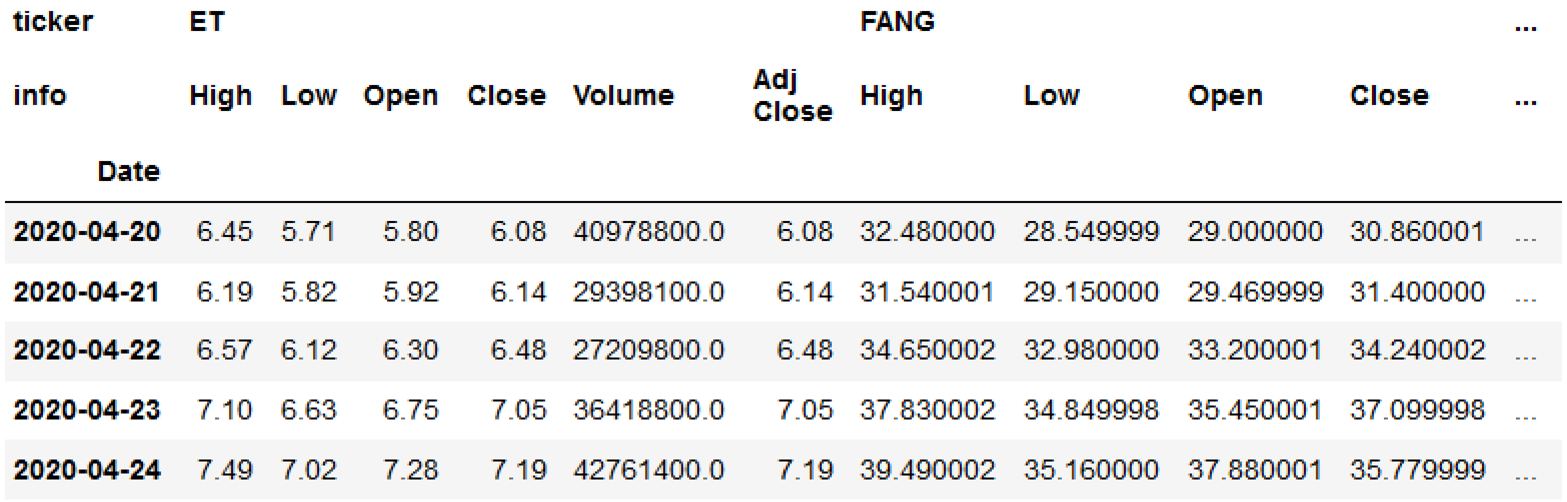
- Filter stocks from the S&P 500 based on users coice
- To perform exploratory data analysis to observe various stock price trend and analyze the stock price behavior of different stocks.
- To develop a simple LSTM based model for predicting the upward/downward trend for a stock using historical stock price data and predict future stock price.
- Use the prediction price from the model to decide which stocks to buy



Energy transfer stock prediction (source: CNN)

Data collection

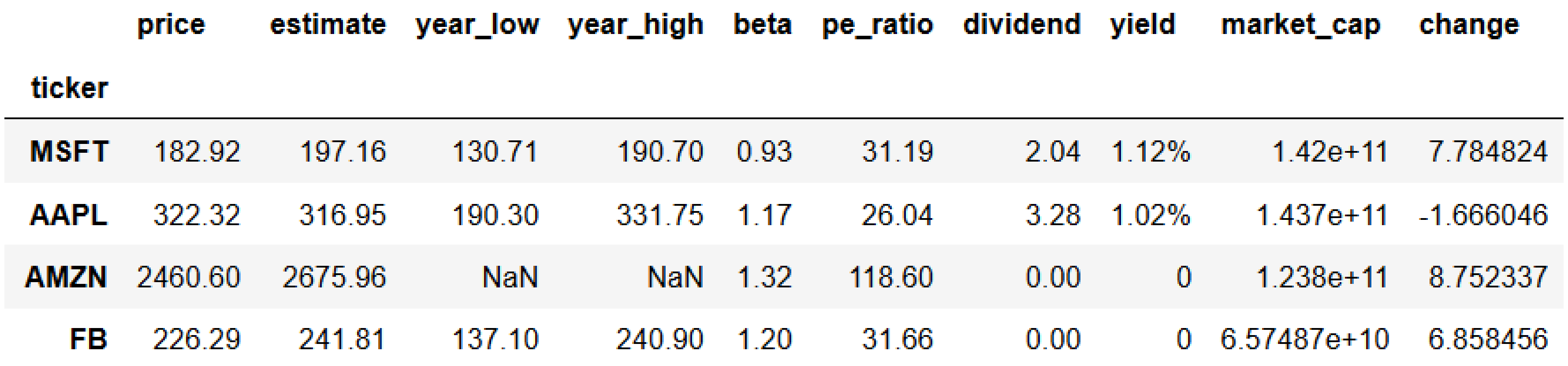
- Used pandas datareader API to scrap stock prices data from ‘yahoo’
- Data was then manipulated to create the following DataFrame where:
  - rows represents time series of date
  - Columns represent a hierarchical table of stock tickers and prices info



ticker	ET						FANG				...
info	High	Low	Open	Close	Volume	Adj Close	High	Low	Open	Close	...
Date											
2020-04-20	6.45	5.71	5.80	6.08	40978800.0	6.08	32.480000	28.549999	29.000000	30.860001	...
2020-04-21	6.19	5.82	5.92	6.14	29398100.0	6.14	31.540001	29.150000	29.469999	31.400000	...
2020-04-22	6.57	6.12	6.30	6.48	27209800.0	6.48	34.650002	32.980000	33.200001	34.240002	...
2020-04-23	7.10	6.63	6.75	7.05	36418800.0	7.05	37.830002	34.849998	35.450001	37.099998	...
2020-04-24	7.49	7.02	7.28	7.19	42761400.0	7.19	39.490002	35.160000	37.880001	35.779999	...

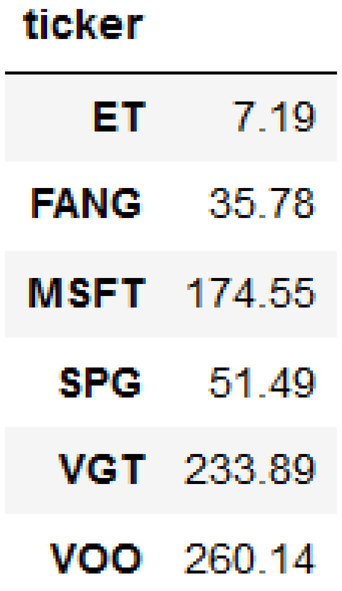
Data cleaning

- First scrap data from yahoo finance
- Populate the following table
- Issues:
    - Market cap includes B for billion, M for million, T from trillion. Convert them to integer
    - Define change using pct_change using price and the wall street estimate
    - Convert objects to float
    - Populate dividend values and replace Nan with zero



	price	estimate	year_low	year_high	beta	pe_ratio	dividend	yield	market_cap	change
ticker										
MSFT	182.92	197.16	130.71	190.70	0.93	31.19	2.04	1.12%	1.42e+11	7.784824
AAPL	322.32	316.95	190.30	331.75	1.17	26.04	3.28	1.02%	1.437e+11	-1.666046
AMZN	2460.60	2675.96	NaN	NaN	1.32	118.60	0.00	0	1.238e+11	8.752337
FB	226.29	241.81	137.10	240.90	1.20	31.66	0.00	0	6.57487e+10	6.858456

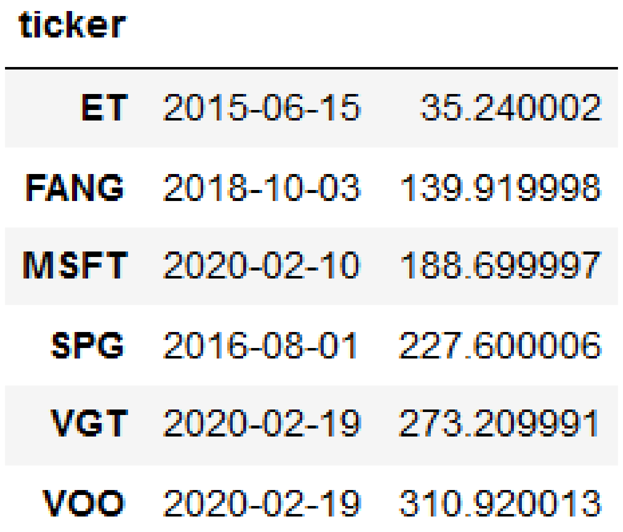
Data analysis



ticker	
ET	7.19
FANG	35.78
MSFT	174.55
SPG	51.49
VGT	233.89
VOO	260.14

Last day’s price

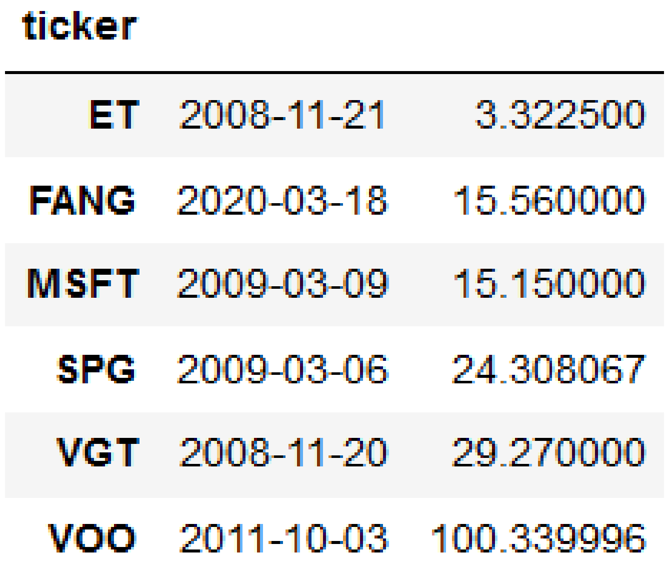
```
pd.merge(pd.DataFrame(my_stocks.xs(key='Close',axis=1
,level='info').idxmax()),
pd.DataFrame(my_stocks.xs(key='Close',axis=1,level='info
').max()),on='ticker')
```



ticker		
ET	2015-06-15	35.240002
FANG	2018-10-03	139.919998
MSFT	2020-02-10	188.699997
SPG	2016-08-01	227.600006
VGT	2020-02-19	273.209991
VOO	2020-02-19	310.920013

All time high

```
pd.merge(pd.DataFrame(my_stocks.xs(key='Close',axis=1,level='info').
idxmin()),
pd.DataFrame(my_stocks.xs(key='Close',axis=1,level='info').min()),on=
'ticker')
```

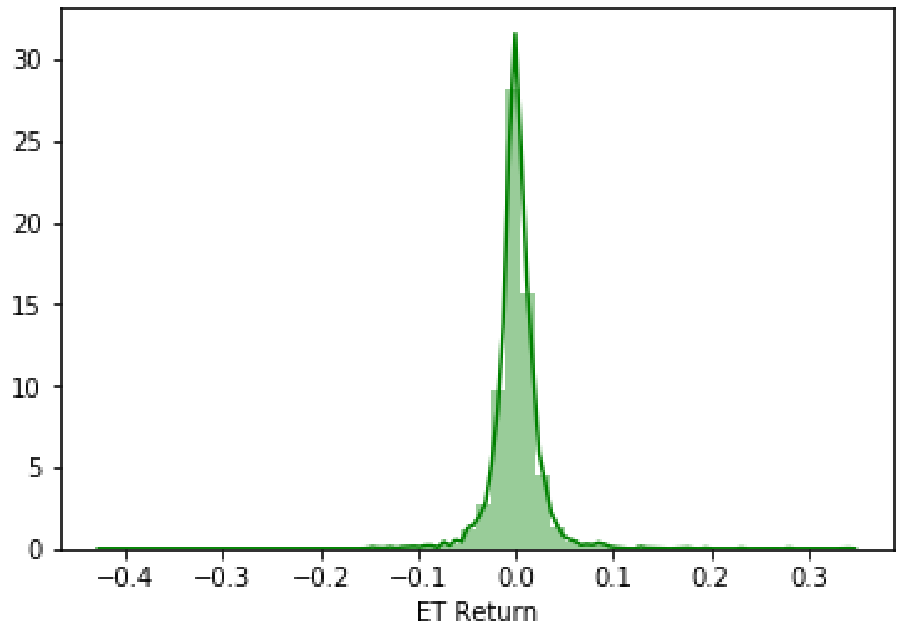


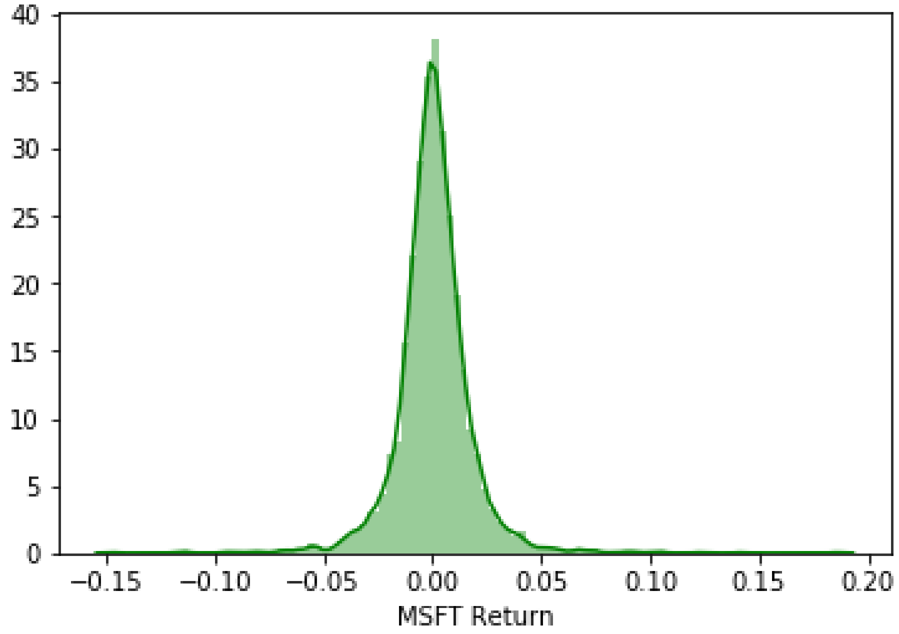
ticker		
ET	2008-11-21	3.322500
FANG	2020-03-18	15.560000
MSFT	2009-03-09	15.150000
SPG	2009-03-06	24.308067
VGT	2008-11-20	29.270000
VOO	2011-10-03	100.339996

All time low

- SPG, FANG, ET are close to their all time low
- MSFT,VOO, VGT are very close to their all time high

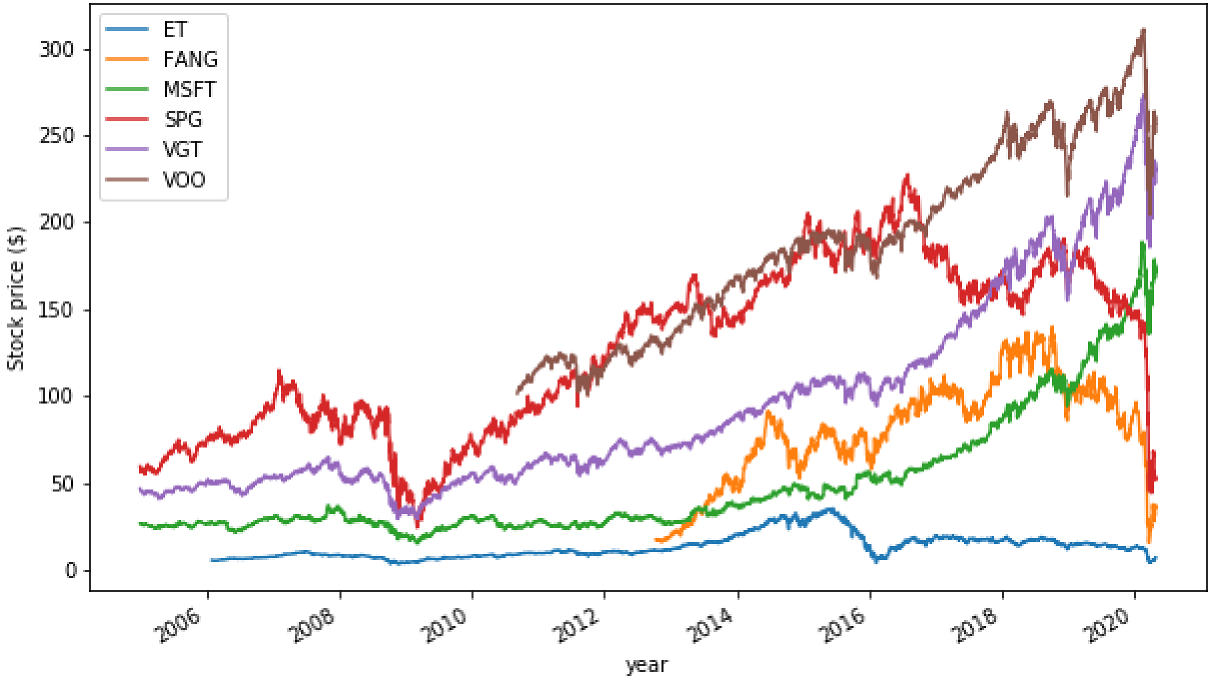
Data analysis



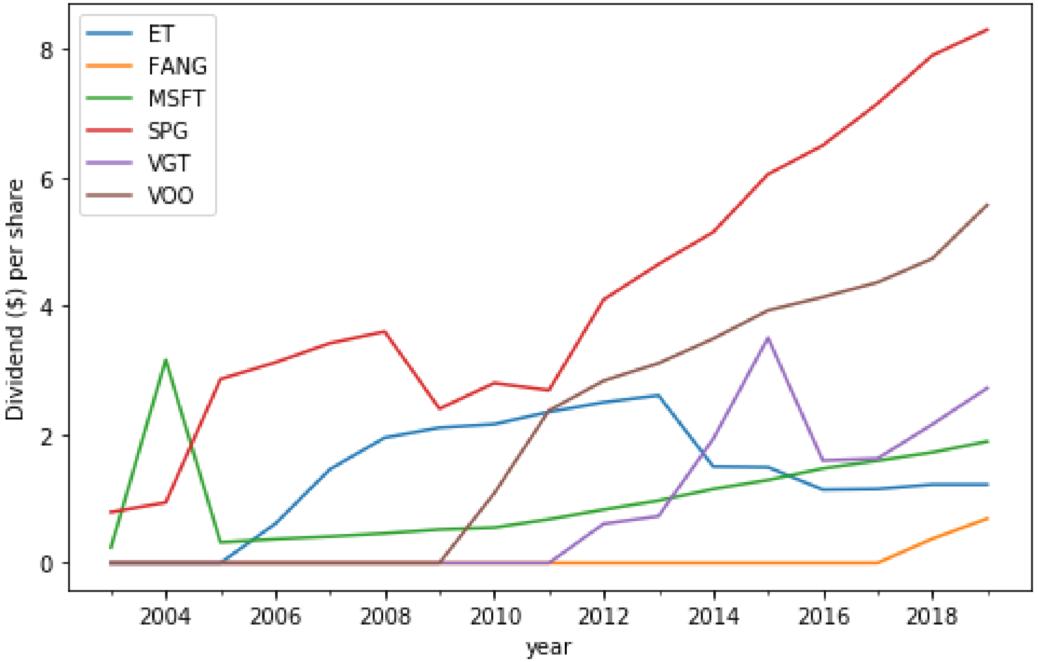


- Distribution plot of the standard deviation of the return (calculated using the percentage change) indicates that ET stock is more fluctuations in prices compared to MSFT

Stock price and dividend over time



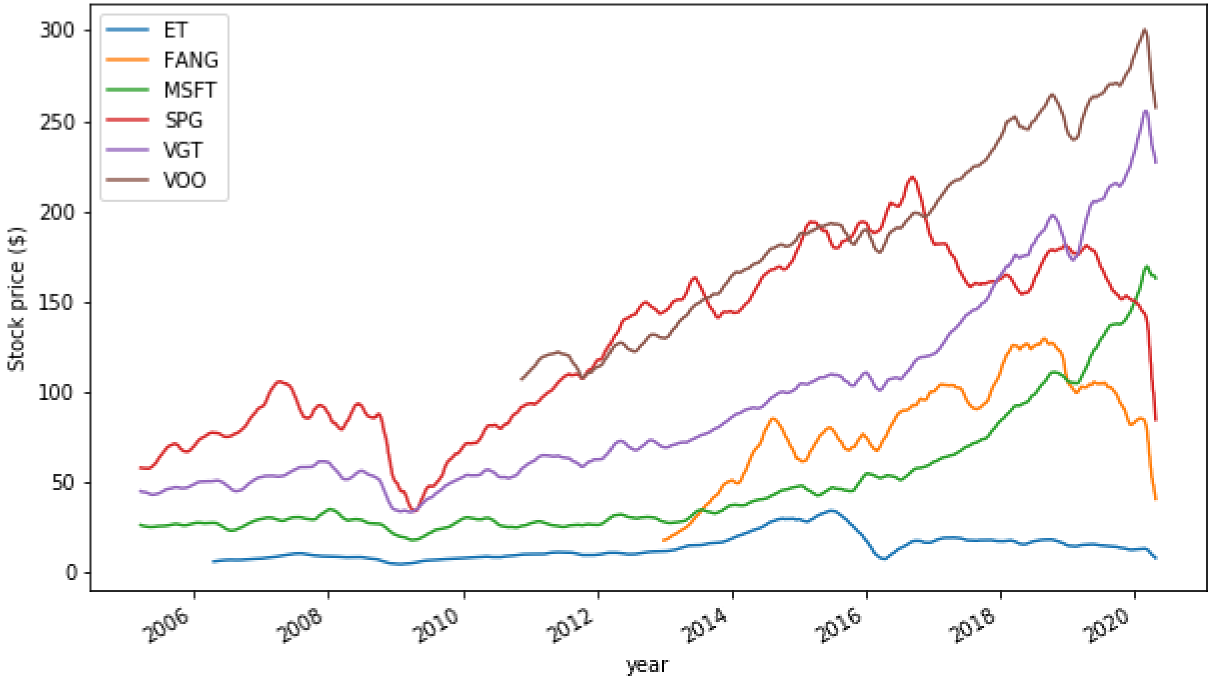
Stock price



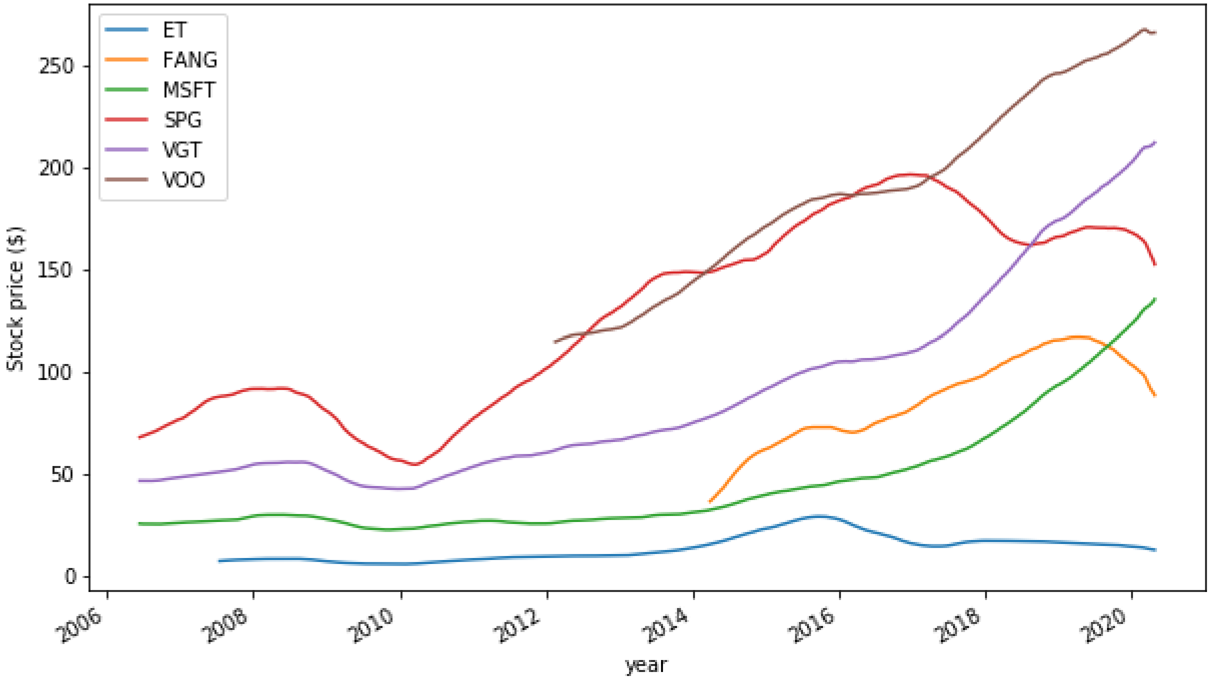
Dividend

- SPG, MSFT, VOO are very stable dividend payers. While ET pays dividend at a decent ratio, but the dividend is not growing

Smoothing the variance of the stock prices



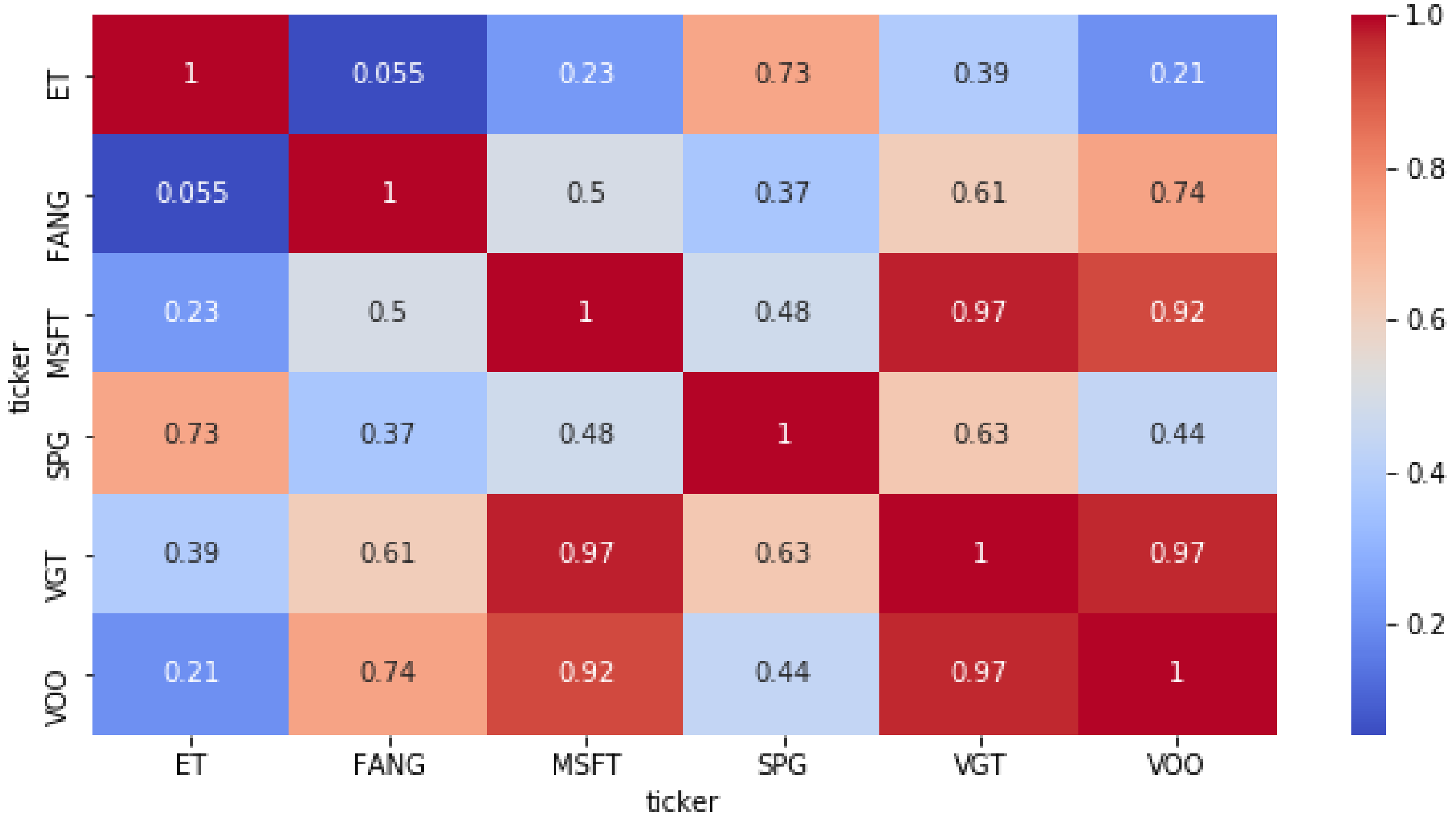
50 day rolling average



365 day rolling average

- Time series of the 50 day and 365 day rolling average gives a high-level idea about the stock type.
- For example, MSFT is a still growing (growth stock)
- Energy transfer pays dividends to the share holders. It's a dividend stock

Correlation between stock price

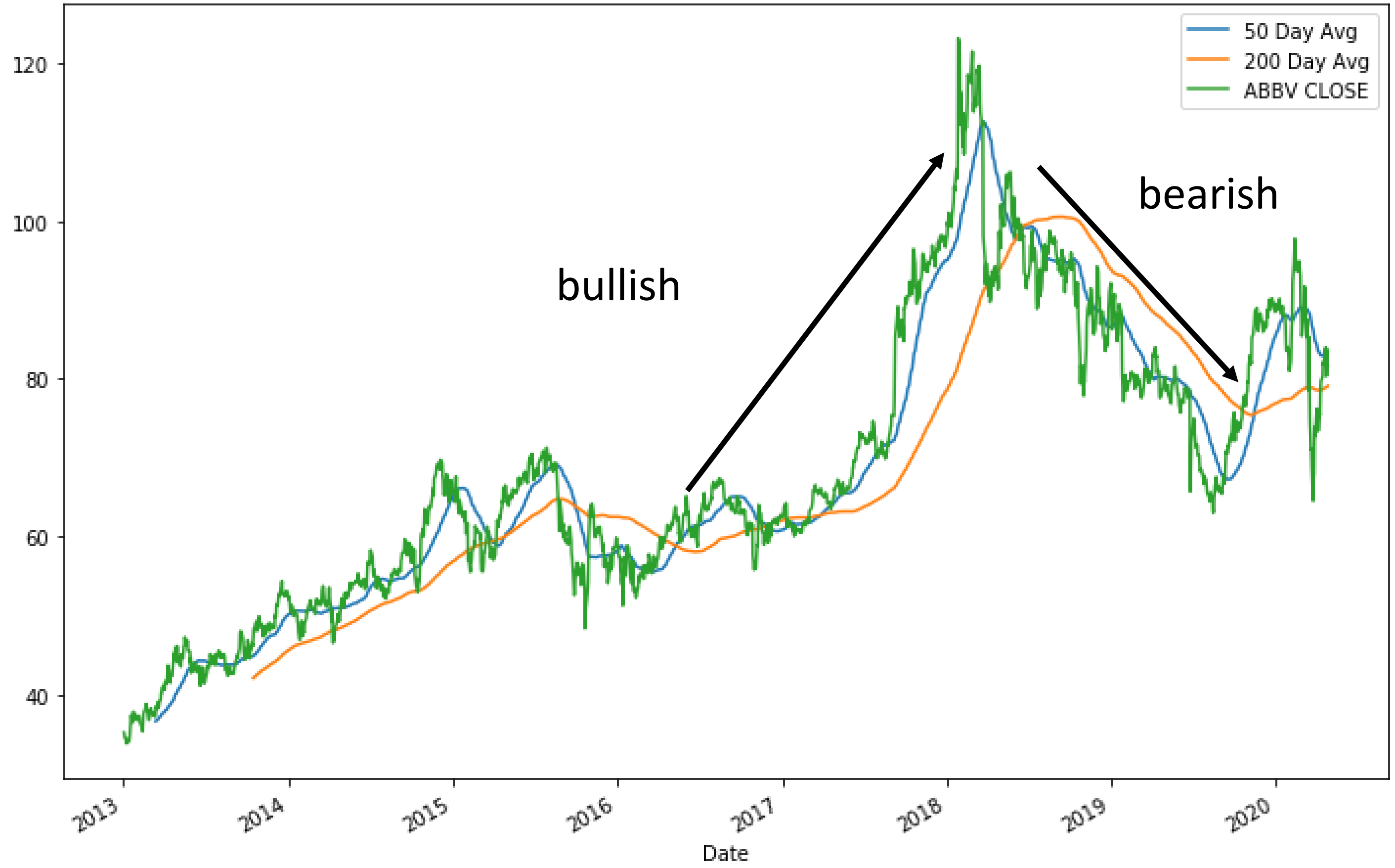


ticker	ET	FANG	MSFT	SPG	VGT	VOO
ET	1	0.055	0.23	0.73	0.39	0.21
FANG	0.055	1	0.5	0.37	0.61	0.74
MSFT	0.23	0.5	1	0.48	0.97	0.92
SPG	0.73	0.37	0.48	1	0.63	0.44
VGT	0.39	0.61	0.97	0.63	1	0.97
VOO	0.21	0.74	0.92	0.44	0.97	1

- Correlation between stocks can be used to identify similar stocks. For example VGT (technology index and VOO (S&P 500 index are heavily correlated))

- To enhance diversity of the portfolio, avoid investing all of your investments in the same bucket

50 Day moving avg vs 200 day moving Avg.



- When 50day moving average cuts the 200day moving average, it is called the golden cross
- 50day moving average cuts the 200day moving average and the slop is positive, it's a bullish signal (buy)
- 50day moving average cuts the 200day moving average and the slop is positive, it's a bearish signal (sell)

The figure above for the ticker 'ABBV' (a pharmaceutical stock) validates the above-mentioned fact

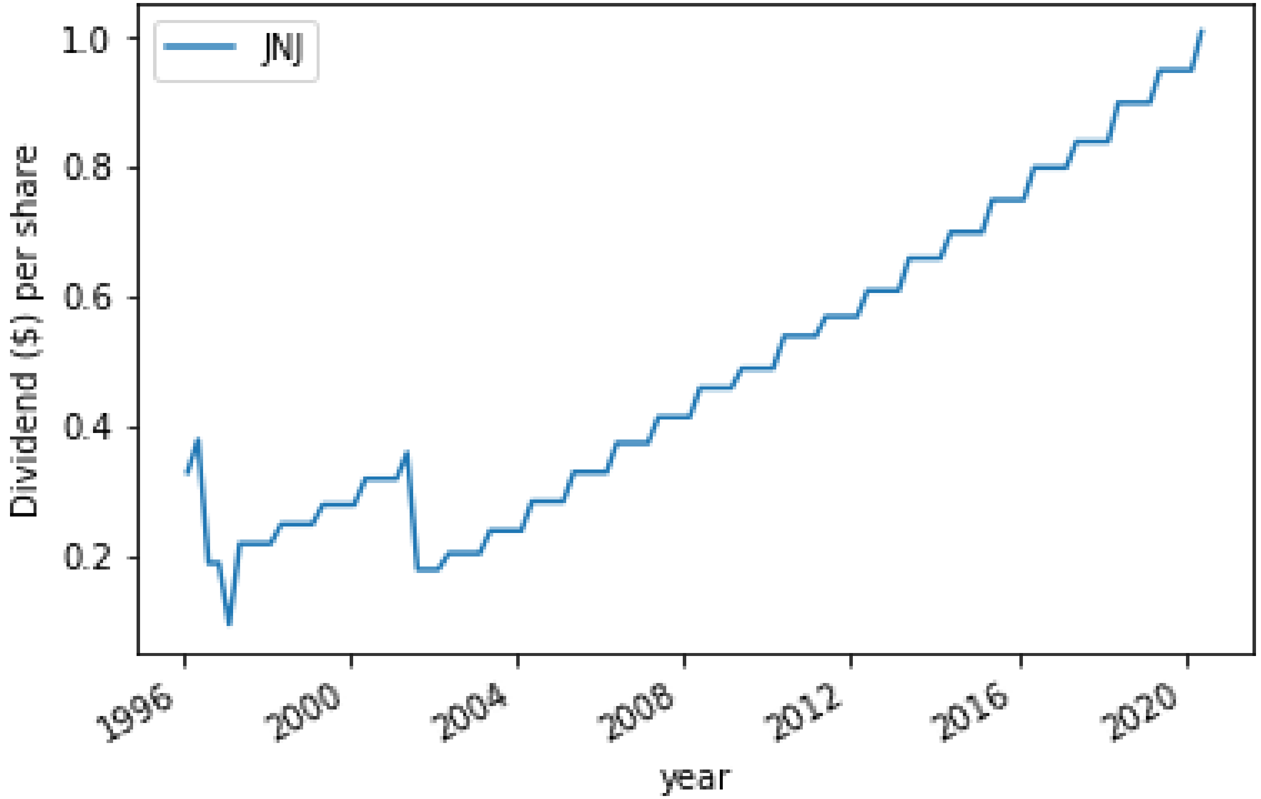
Filtering

- Used filtering based on users choice to filter stocks to buy.
- For example, see the following filtering: here user wants tofilter stocks that have market cap over 10B, beta values less than 1 (less risky stocks), pct change greater than 10 (greater reward), pays dividend greater than 3%, and pe ratio less than 30 (not overpriced)

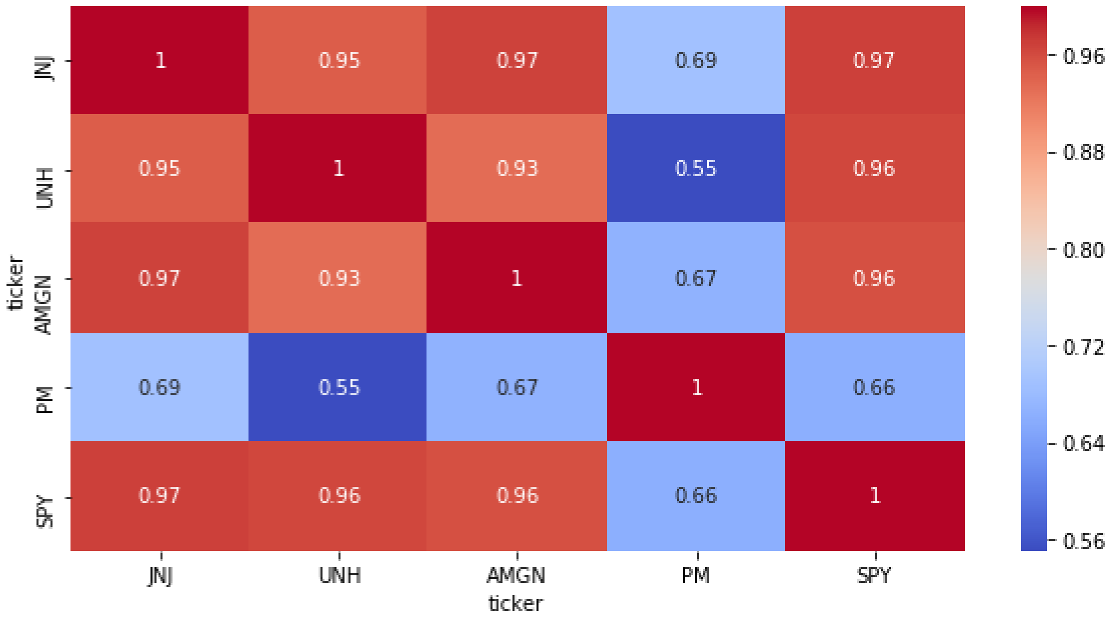
```
filter = stocks_filter[(stocks_filter.market_cap> 10000000000) & (stocks_filter.beta
< 1)  & (stocks_filter.change >10) & (stocks_filter.pe_ratio <30) &
(stocks_filter.dividend >3)]
```

- The filtering criterion resulted with the following 4 tickers JNJ, AMGN, PM, UNH
- Notice that 3 out of 4 stocks are pharmaceutical, wall street expects higher return from pharma stocks in future as response to covid 19

Dividend (example JNJ)



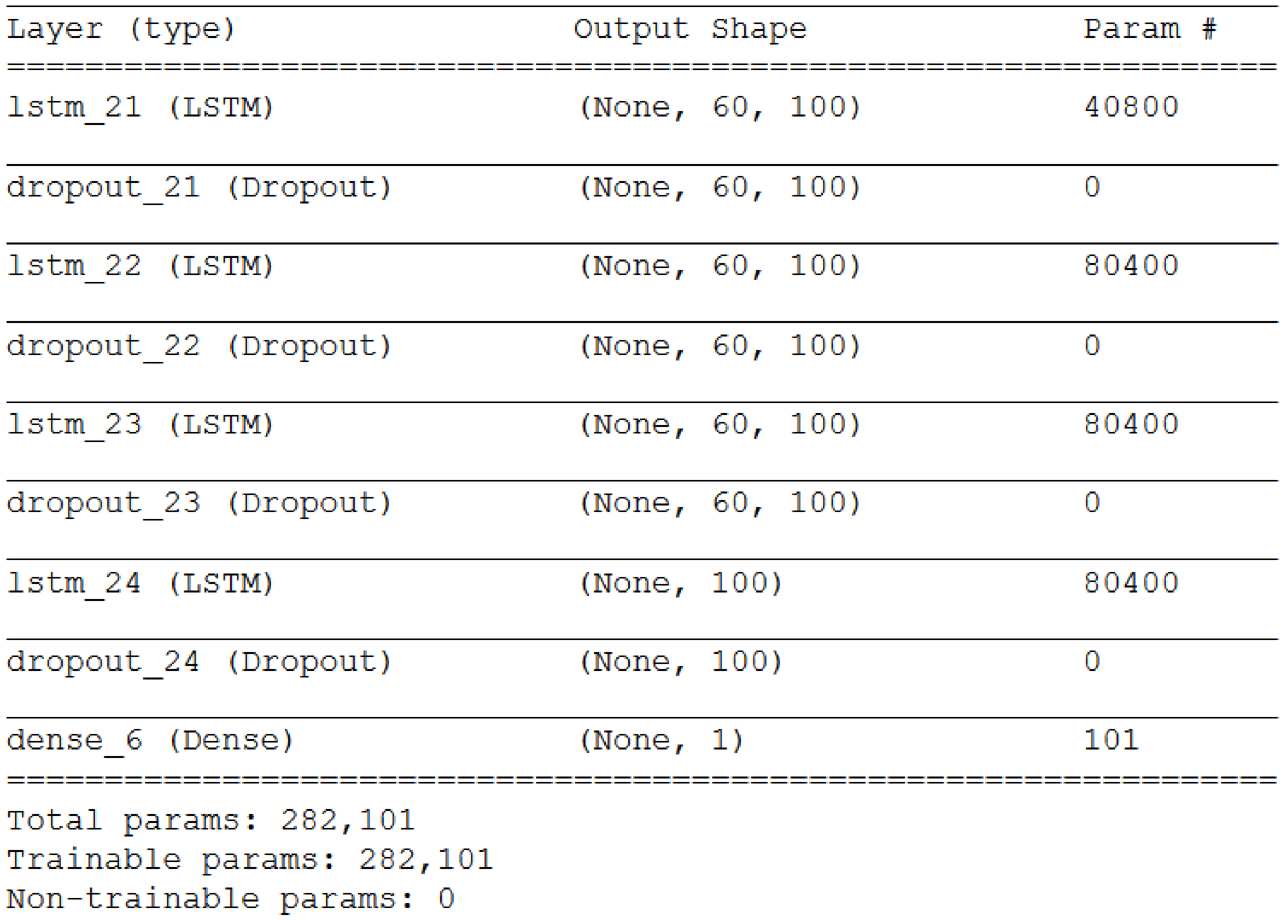
Correlation between stock price



ticker	JNJ	UNH	AMGN	PM	SPY
JNJ	1	0.95	0.97	0.69	0.97
UNH	0.95	1	0.93	0.55	0.96
AMGN	0.97	0.93	1	0.67	0.96
PM	0.69	0.55	0.67	1	0.66
SPY	0.97	0.96	0.96	0.66	1

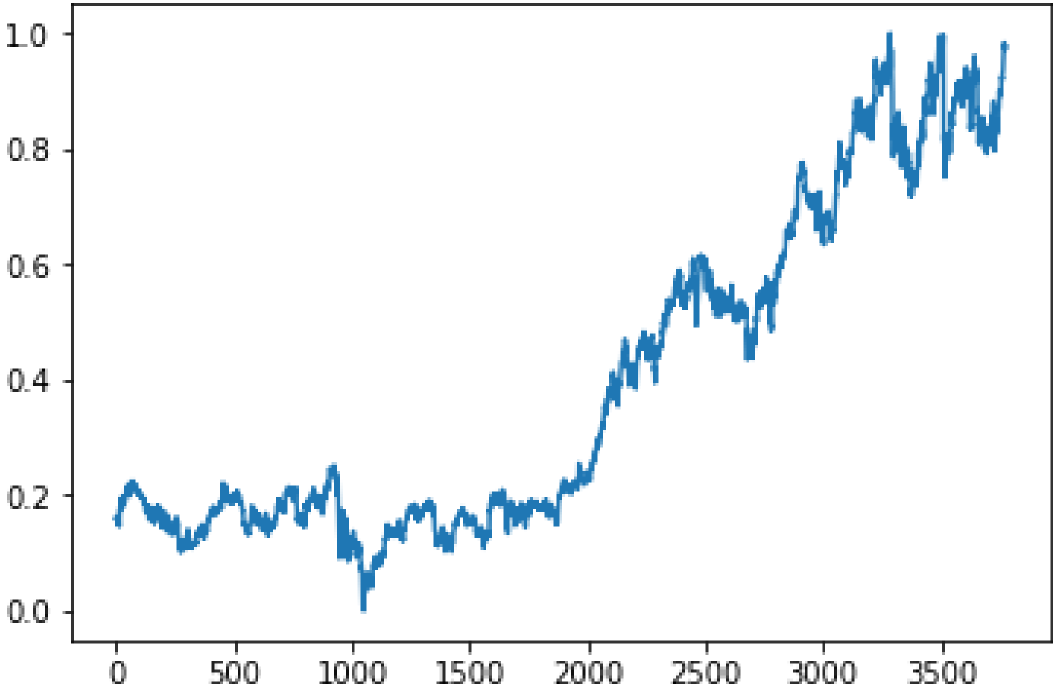
- Correlation between stocks can be used to identify similar stocks. For example SPY (S&P500 index P 500 index are heavily correlated)) and pharma stocks are heavily correlated

Summary of the LSTM model and parameters

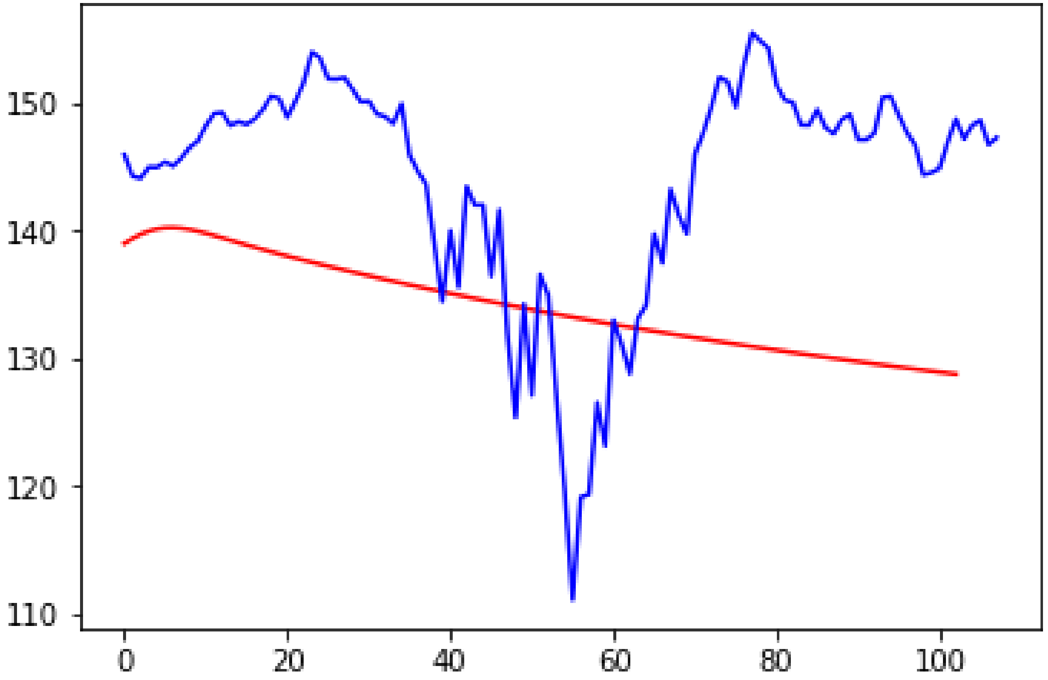


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_21 (LSTM)               (None, 60, 100)           40800     
_________________________________________________________________
dropout_21 (Dropout)         (None, 60, 100)           0         
_________________________________________________________________
lstm_22 (LSTM)               (None, 60, 100)           80400     
_________________________________________________________________
dropout_22 (Dropout)         (None, 60, 100)           0         
_________________________________________________________________
lstm_23 (LSTM)               (None, 60, 100)           80400     
_________________________________________________________________
dropout_23 (Dropout)         (None, 60, 100)           0         
_________________________________________________________________
lstm_24 (LSTM)               (None, 100)               80400     
_________________________________________________________________
dropout_24 (Dropout)         (None, 100)               0         
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 101       
=================================================================
Total params: 282,101
Trainable params: 282,101
Non-trainable params: 0
```

JNJ

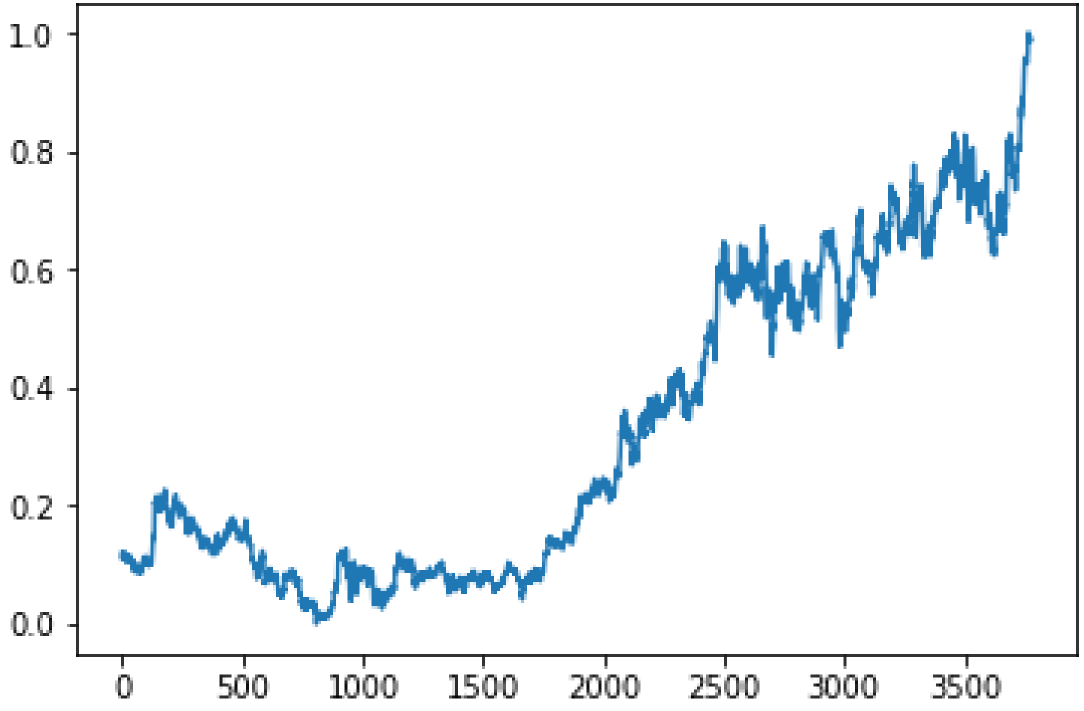


Time series data from 2005 to 2019
(training data)

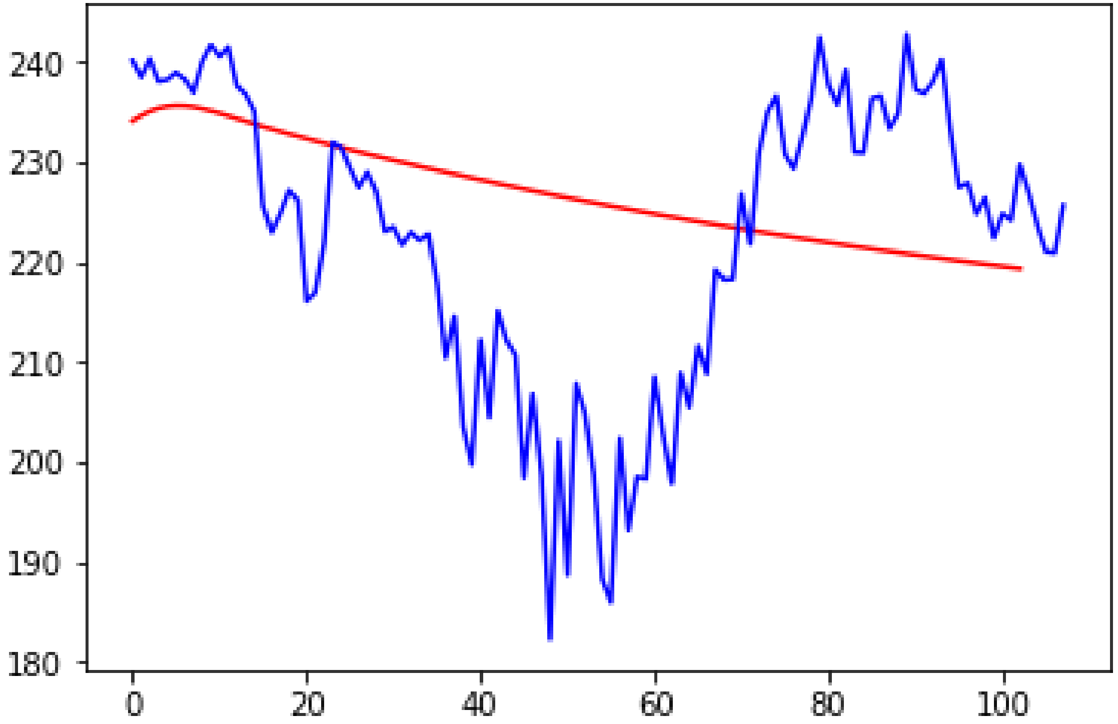


Time series data from for the 5 months of 2020(test data). Red line shows the predicted stock prices using LSTM model

AMGN

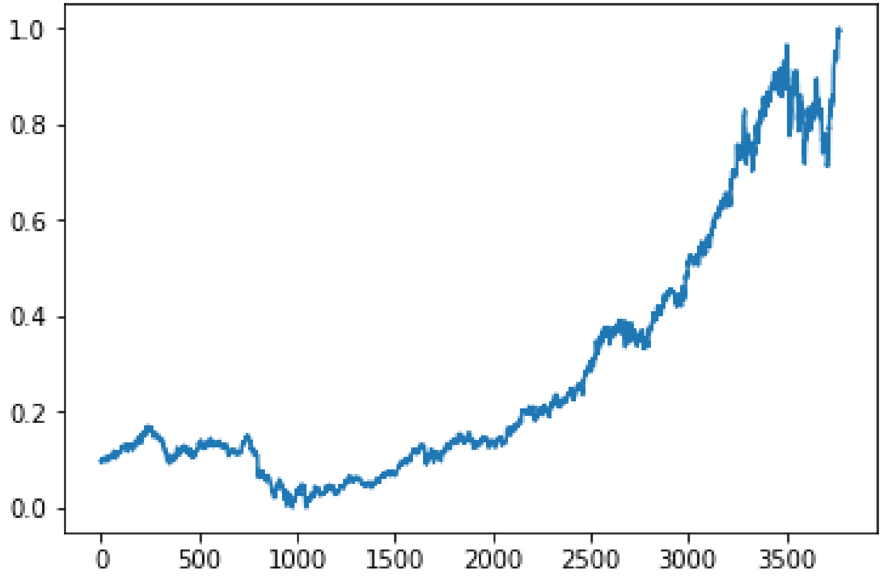


Time series data from 2005 to 2019 (training data)

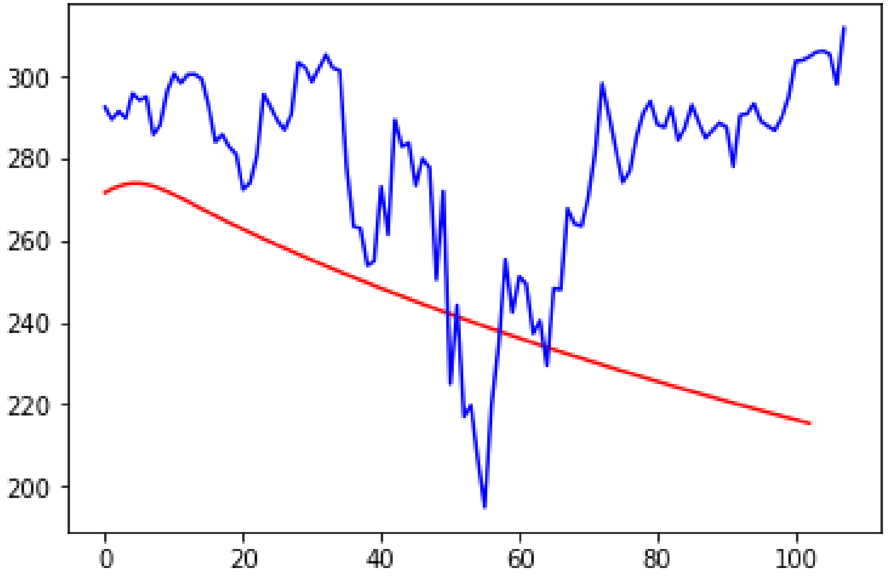


Time series data from for the 5 months of 2020(test data). Red line shows the predicted stock prices using LSTM model

UNH

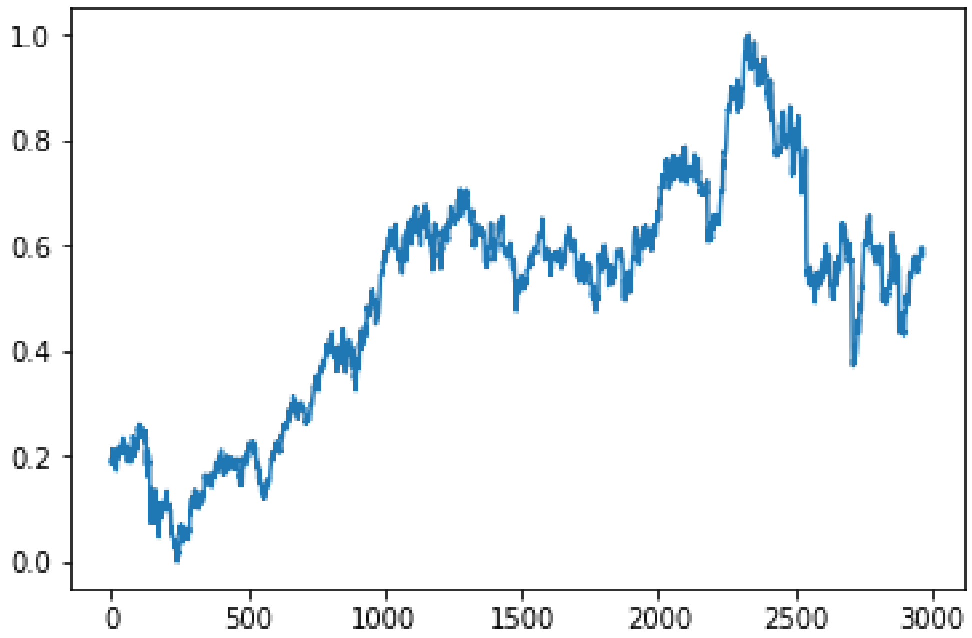


Time series data from 2005 to 2019 (training data)

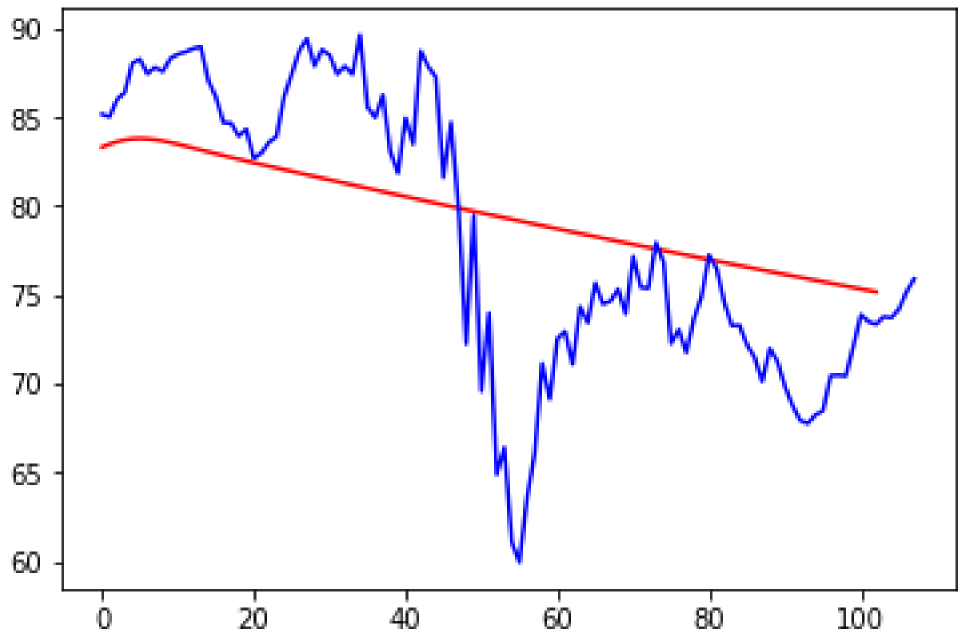


Time series data from for the 5 months of 2020(test data). Red line shows the predicted stock prices using LSTM model

PM

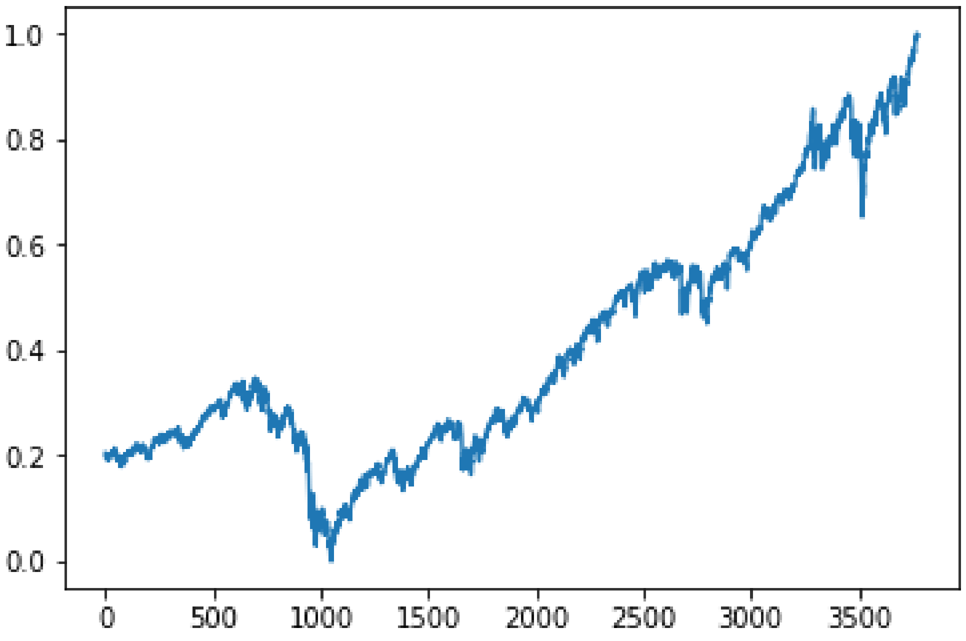


Time series data from 2005 to 2019 (training data)

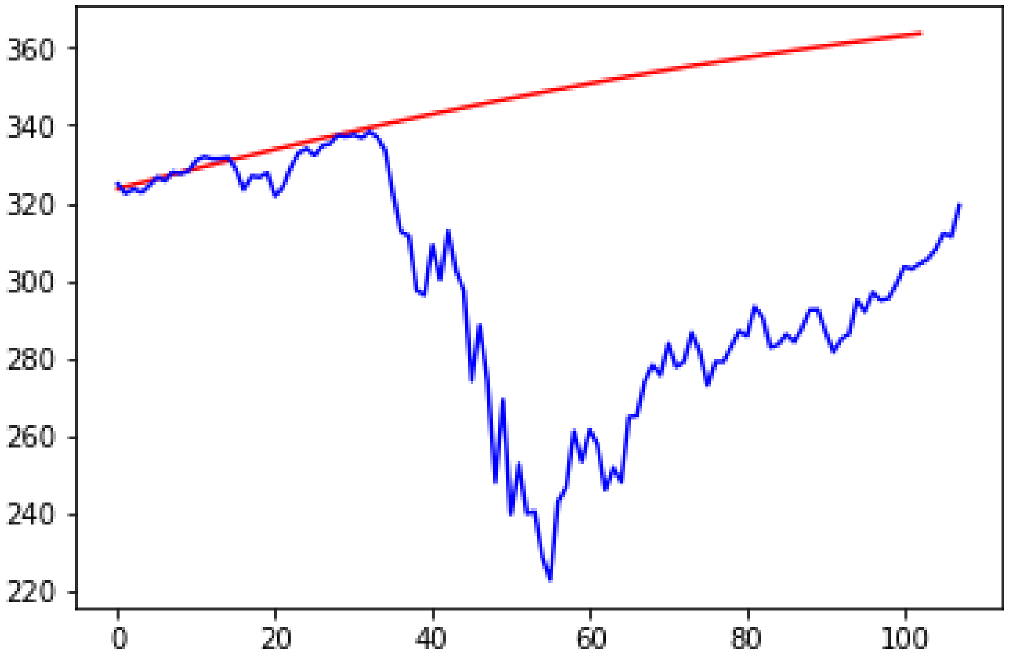


Time series data from for the 5 months of 2020(test data). Red line shows the predicted stock prices using LSTM model

SPY



Time series data from 2005 to 2019 (training data)



Time series data from for the 5 months of 2020(test data). Red line shows the predicted stock prices using LSTM model

Table 1: prediction analysis

For simplicity I assumed, equal investments on the filtered stocks

Ticker	price on Dec 31	Model (5 months predicted)	prediction MSE	actual	model return	model return (individual)	actual return (individual)	Actual return
JNJ	145.87	128.78	13.43	146.99	-14.8%	-11.7%	0.77%	-2.701%
AMGN	241.07	219.30663	16.01	222.74		-9.0%	-7.60%	
PM	85.09	75.18	6.08	77.1		-11.6%	-9.39%	
UNH	293.98	215.39	44	310.75		-26.7%	5.70%	
SPY	321.86	363.61	62.72	320.68	13.0%	13.0%	-0.37%	-0.37%

Analysis

Model return	-14.8%
Actual return (filtered stocks)	-2.7%
model SPY return	13.0%
Actual SPY return	-0.4%

- Model return suggests significant loss compared to SPY. So suggestion is to buy SPY (SP 500 index)

- From the actual data, we can see that although both returns are negative, SPY loss is less compared to the filtered stock.

Future works

- Add more features in prediction, seasonality, holiday sales, quarterly information and so on.
- Recently I came to know about facebook prophet library for time series which integrates new features easily. It would be nice to play with that.
- Running neural networks on my computer is a pain. It takes ages to run 5-10 epochs. Further hyperparameter tuning, grid search, more epochs would increase the RMSE of the model.
- Do the linear optimization to find the value to weight parameters.