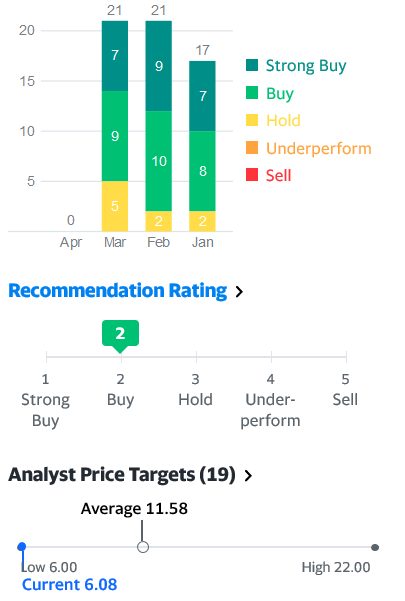
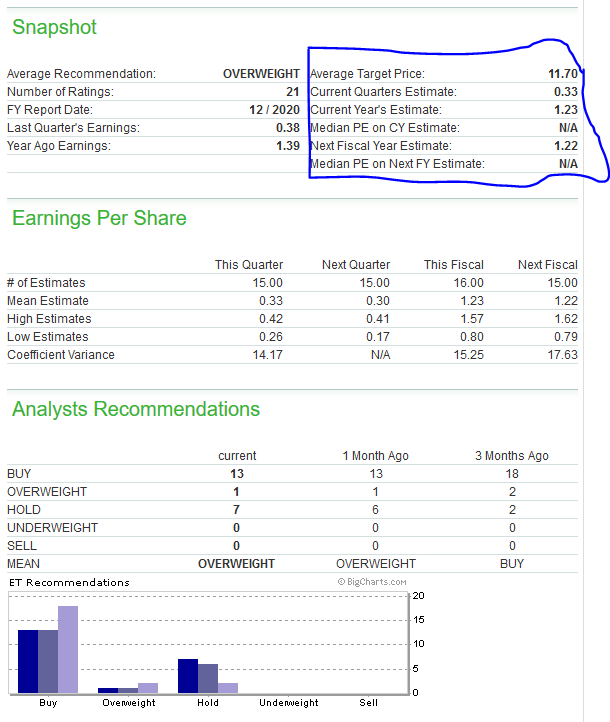
**Project: Stock price forecasting using LSTM/ARIMA and investment decesion**

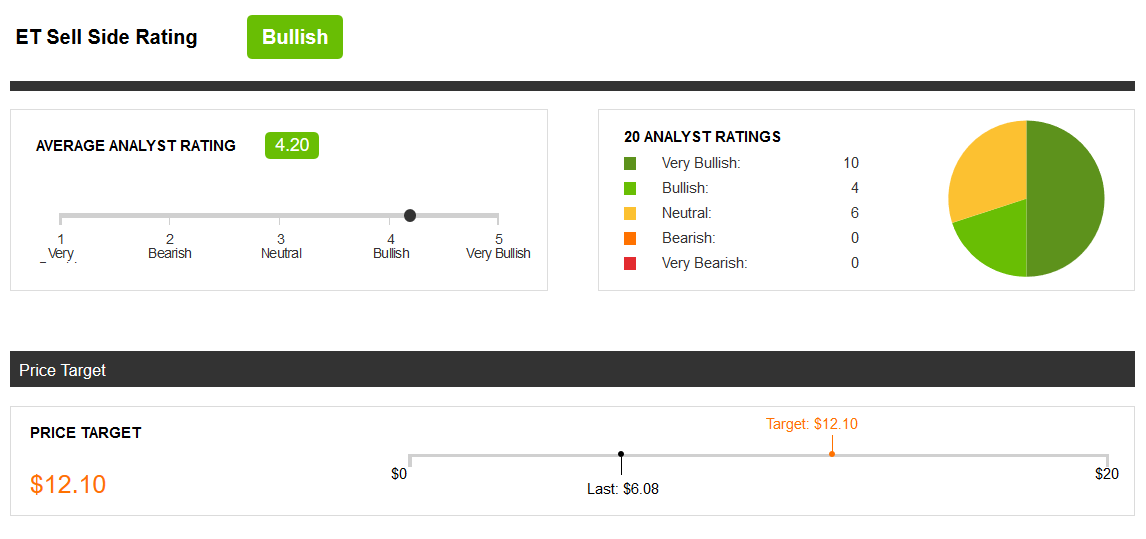
**Problem:** I have been investing in the stock market for a few months starting from last August. The recent fluctuations in the market have made new investors like me vulnerable. I have invested using the forecasts and 1-year estimates collected from CNN, Yahoo, Market Watch, seeking alpha, NBC websites. For example, see the figure below for the stock price of energy transfer taken from CNN finance.



The figure shows the last 12 months of the stock price for this company (current price is $6.07) and the one-year median forecast of $12. The high and low represents the confidence interval. This forecast is made by stock market analysts.  Based on this forecast, analysts think that there would be a 97.4% increase in this stock price in the next 12 months. Similar forecasts are available on other websites as well.  The estimates change in every quarter due to earnings reports or due to any special situations like the current coronavirus fluctuations or any internal company issues.

Similar stock prediction from other sources are also available. For example, see the yahoo estimate (left), market watch estimate (right), seeking alpha (below).



So, the CNN estimate is $12, yahoo estimate is $11.58 and the Marketwatch estimate is $11.70 and seeking alphas is $12.10. The average of estimates in 11.84. The current price of the stock is $6.1. This indicates that if I invest in this stock there is a strong chance of making a 94% return on investment in one year!

The problem is:

1. The underlying rationale for these predictions is not known.
2. There are thousands of stocks available. It’s not possible to manually look for potential stocks with high yield estimates.

For example, I used the above intuitions to invest in two oil stocks. And currently at 40% down on my investments. Why? Because my profile is not diversified enough. For example, due to coronavirus effect, retail and oil were hurt severely compared to tech stocks.

In this project, I aim 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. First, I filter stocks based on users input and then use forecasting algorithm to predict the stock prices. Then I would tabulate the results and compare it with the analyst estimates from various other sources. Finally I would I would like to decide using the filtered stocks and S&P500 index that whether it is a good choice to invest in individual stocks or just invest in the index.

Finally, the investment decision can be made based on a combination of the return of investment and the diversity of the portfolio. We can use linear optimization technique for solving this problem

**User’s/client:** Financial institutes, stock investors. But the primary user would be new/small investors like me who want to make a data-driven decision using a simple approach rather than going through the numbers and balance sheets used by the financial analysts which can be daunting.

**Dataset:** I used yahoo stock API to scrap stock price for the stocks.

- Used pandas datareader API to scrap stock prices data from ‘yahoo’.  
- Data was then manipulated to create the following DataFrame where:

- Rows represent time series of date.

- Columns represent a hierarchical table of stock tickers and prices info.

I scrapped the stocks with following tickers:

MSFT: Microsoft

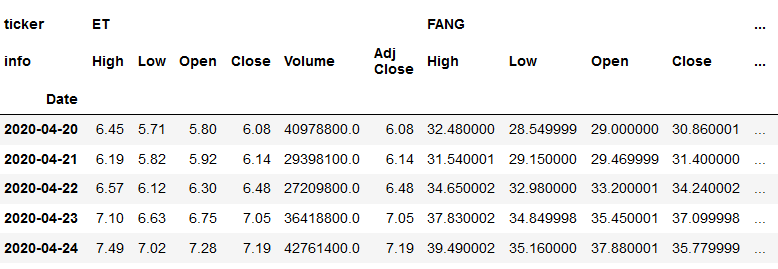
ET: Energy transfers

SPG: Simon properties group

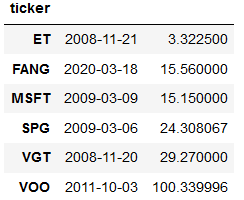
FANG: Diamondback energy

VGT: Vanguard technology ETF

VOO: Vanguard S&P500 ETF



**Data analysis:**

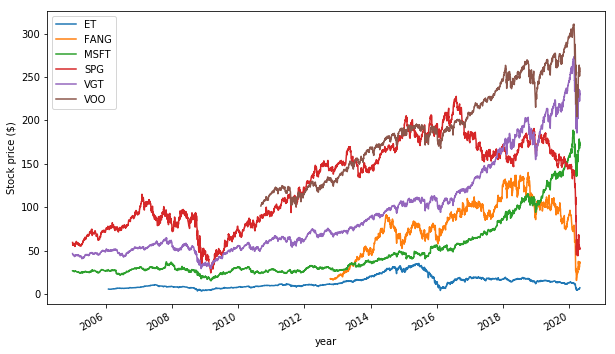
First, lets see how the stocks are performing currently compared to their all-time high and lows.

All time low

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Current price | All time low price | All the high price |

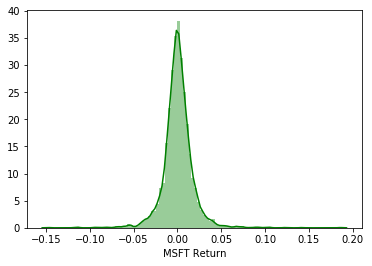
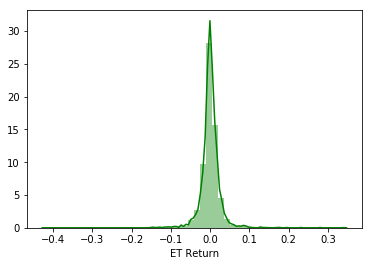
In the table above, we show the current, all the high and all-time low-price side by side for the stocks of our analysis. SPG, FANG, ET are close to their all-time low. Again, MSFT,VOO, VGT are very close to their all-time high.

This can be further visualized in the chart below where I have plotted the time series of the stock prices.

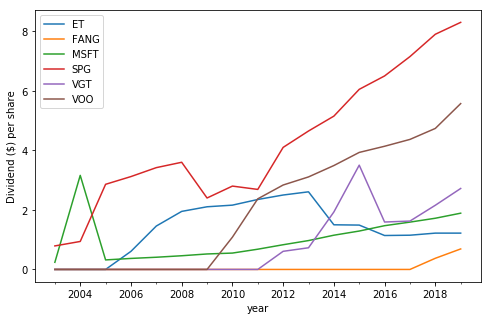


It can be seen that Microsoft, VGT and VOO prices grew considerably in the last 15 years. FANG, and ET stocks declined considerably recently especially due to lack of oil demand.

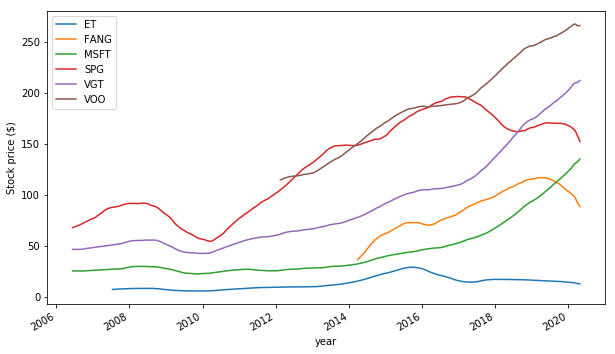
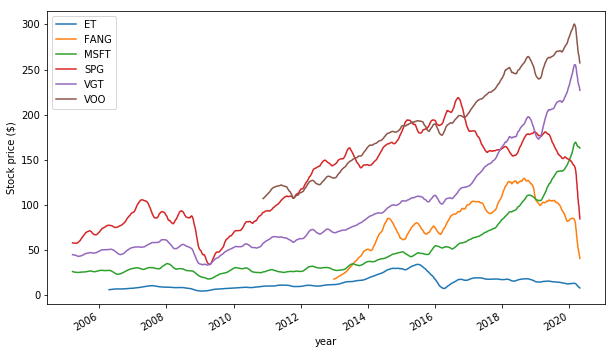
Next, I checked the distribution plot of the standard deviation of the return (calculated using the percentage change) indicates that ET stock is more fluctuations in prices compared to MSFT.



Next, I check the yearly paid dividends by those stocks. SPG, MSFT, VOO are very stable dividend payers. While ET pays dividend at a decent ratio, but the dividend is not growing.



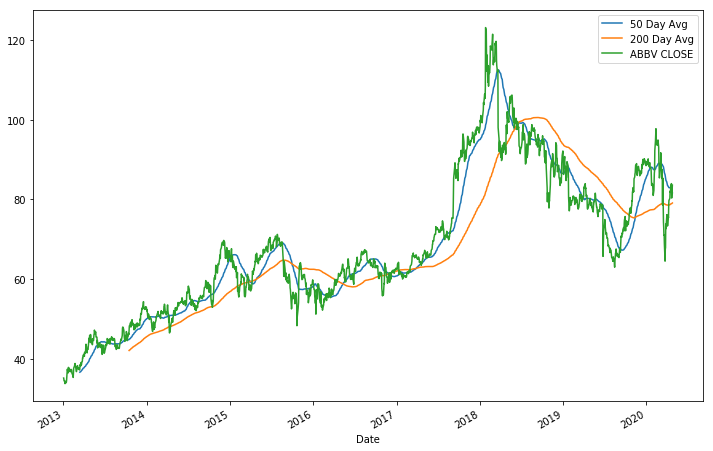
Next, we plot the stock price using the 50 day and 365 day rolling average. Time series of the 200/365 day rolling average gives a high-level idea about the stock type. For example, MSFT is a still growing (growth stock). Energy transfer pays dividends to the shareholders. It’s a dividend stock.

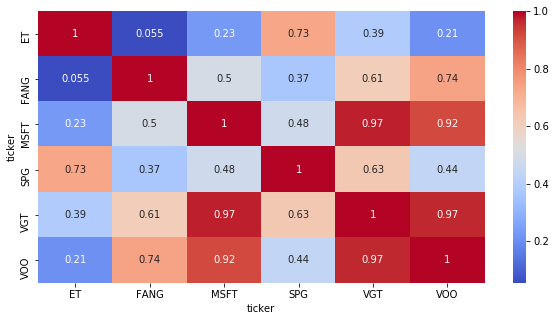


365 day rolling average

50 day rolling average

Next, we check the golden cross behavior i.e., 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





Next, we check the Correlation between stock price. 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

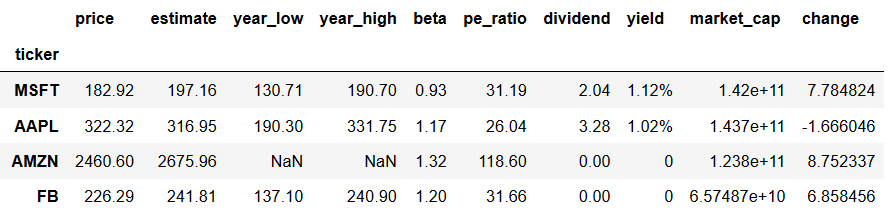
**Filtering the stocks:**

With the analysis at hand, first I scrap the current values of various features of the S&P 500 stocks. Features include price, 52 week high, 52 week low, wall street estimate, market cap, dividend per share, dividend yield, pe ratio, beta, and pct\_change.

The following steps were used to data collection and 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

Following table is a snapshot of the collected data. The data collection process is very time consuming. The loop runs for 503 times of the S&P500 stocks and then scraps individual stocks and their feature values.



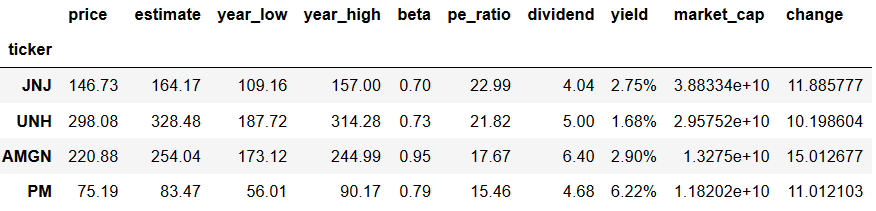
Filtering stocks:

Then I use filtering based on users choice to filter stocks to buy.

For example, see the following filtering:

* user wants to filter 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)

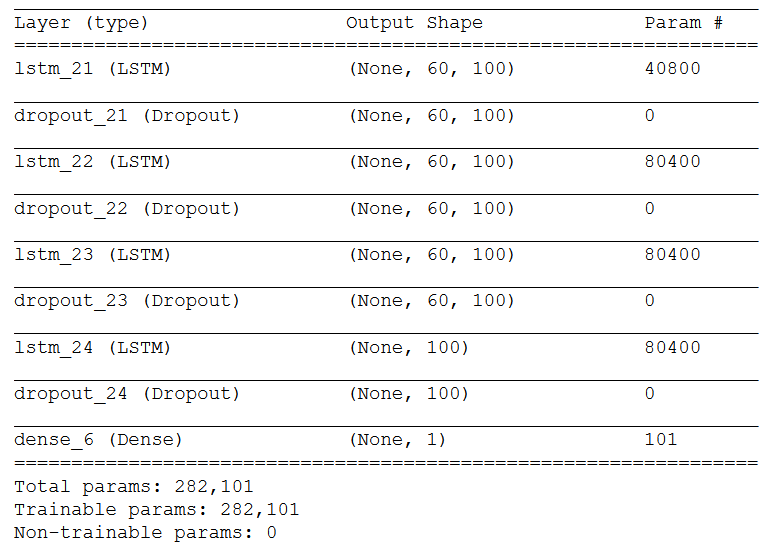
The filtering criterion resulted with the following 4 tickers JNJ, AMGN, PM, UNH. Notice that 3 out of 4 stocks are pharmaceutical, wall street expects higher return from pharma stocks in future as response to covid 19



**Predicting stock prices using LSTM:**

Stock prices are time series data where current prices are influenced by the prices on the previous day. One difference between a classical regression problem with a time series data is that in a time series the values on a current time depends on the values of the previous days. A recurrent neural network can process this sort of sequential data is designed in such way that the output at a current time step can be predicted from its previous values.

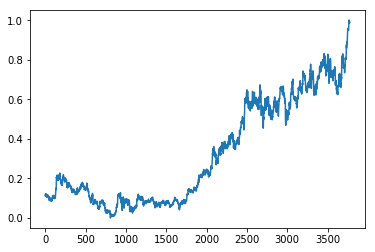
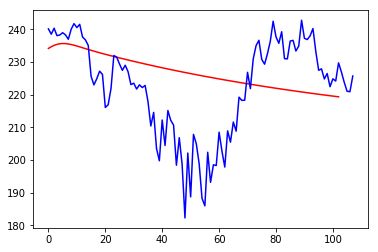
Following are the chosen LSTM parameters. For details on modeling using LSTM please visit the blog by Karpathy [here](http://karpathy.github.io/2015/05/21/rnn-effectiveness/)



**Results from the prediction:**

Training data for SPY, AMGN, JNJ and UNH includes 15 years of time series ( Jan 2005 to Dec 2019). Training data for PM includes 12 years of time series ( Jan 2008 to Dec 2019).

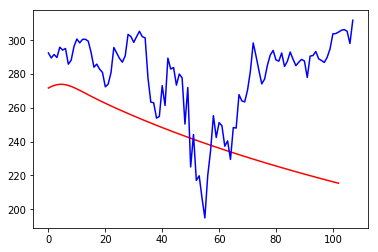
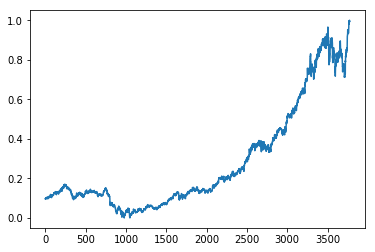
AMGN



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

Time series data from 2005 to 2019 (training data)

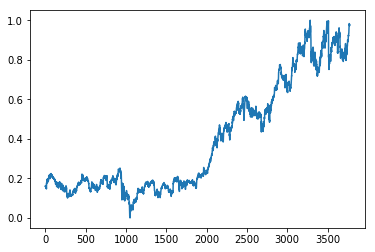
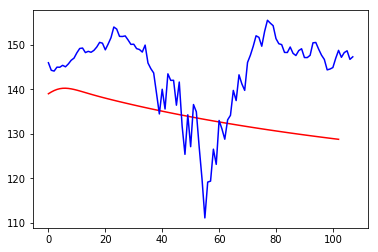
UNH:



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

Time series data from 2005 to 2019 (training data)

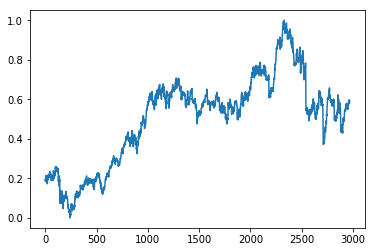
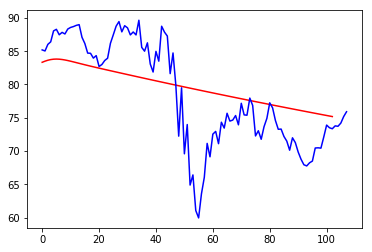
JNJ:

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

Time series data from 2005 to 2019 (training data)

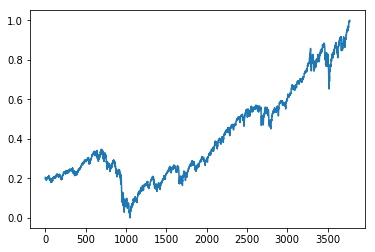
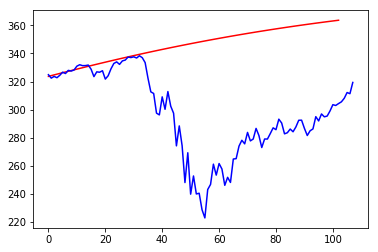
PM:

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

Time series data from 2005 to 2019 (training data)

SPY:

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

Time series data from 2005 to 2019 (training data)

**Analysis:**

Based on the aforementioned technique, I now do a very simple analysis to decide whether I should have bought the S&P500 index or the 4 stocks that I filtered.

Table 1 (prediction analysis) is populated with the actual and predicted prices and returns. 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% |

* 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.

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

Thus a simple conclusion leads to the fact that the user should buy SPY index compared to the 4 stocks he/she filtered. It is needed to mention that the return on investment may lead to an opposite result if the weights on individual stocks are chosen carefully suing linear optimization and proper design of the constraints.

**Summary and future Works:**

This approach is helpful to filter good stocks from S&P500 and heavily automates the stock analysis process. Further, the forecasting gives a high-level overview of the trend of the stocks. To make a product from the project, I need to 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. I need to do the linear optimization to find the value to weight parameters. However, It needs to be mentioned here that the predicted RMSE is heavily impacted by the significant dip in March and April of 2020 in the test data. I intentionally choose this window as test data to check how the model is going to perform on such an extreme case. In practice, this dip needs to be modeled separately in order to understand and predict the regular trend.