ISyE 6740 – Spring 2022
Final Report

Team Member Names: Siddharth Gudiduri

Project Title: Stock Market Investment

Please include (at least) the following sections.

Problem Statement

The purpose of this project is to **maximize** expected returns on investment made on a portfolio, while **minimizing** risk of that portfolio. We are going to base our problem on **Modern Portfolio Theory**, a mathematical framework by Harry Markowitz. Key idea here is that asset's risk and returns should not be assessed by itself (stock in this case) but how it contributes to portfolio's overall returns. In this problem, we quantify risk by variability of stock's price over a fixed time(volatility). This project's **novelty** lies in adding additional constraints, like expected minimum return, ability to select or exclude stocks and to what degree.

Formulating above problem, let's define variables, constraints, and objective function.

Definitions:

B = investment budget
ret = minimum return defined by user
x_i = amount invested in stock i
r_i = expected return of stock i
Q_{ij} = covariance of returns of stocks i and j
m_i = minimum dollar amount for stock i
$\Sigma_i r_i x_i$ or $\boldsymbol{r^T}X$ = Return on Investment
$\Sigma_i \Sigma_j\ Q_{ij}\ x_i x_j$ or $\boldsymbol{x^T}Qx$ = Variance (Volatility)

Variables:

x_i = amount invested in stock i

Constraints:

$\Sigma_i x_i \leq B$ amount invested in stock i
$x_i \geq 0$ for all stocks i

Objective Function:

Maximize $\Sigma_i r_i x_i - \gamma * \Sigma_i \Sigma_j Q_{ij} x_i x_j$

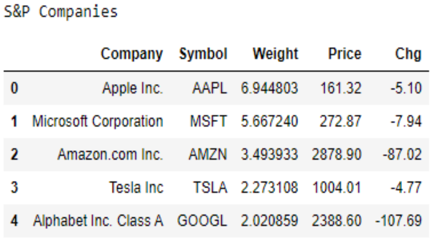
Additional Constraints:

$\Sigma_i r_i x_i \geq$ ret (user defined, minimum return value)

x_i>=.1 or x_i<=.99 (to include stock at min and max degree)
x_j<=0.0(to exclude stock j, otherwise considered)

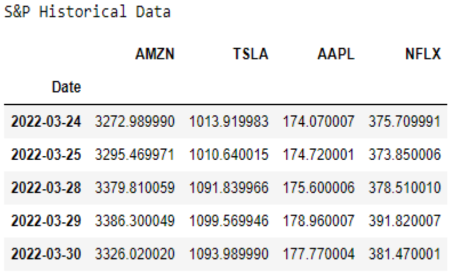
Data Source

Dataset was obtained by scraping https://www.slickcharts.com/sp500 and using yahoo finance API. First dataset included current S&P companies and their current metadata. Second dataset included all S&P companies historical price information since 03/2020. Below examples of some companies in datasets



S&P Companies

	Company	Symbol	Weight	Price	Chg
0	Apple Inc.	AAPL	6.944803	161.32	-5.10
1	Microsoft Corporation	MSFT	5.667240	272.87	-7.94
2	Amazon.com Inc.	AMZN	3.493933	2878.90	-87.02
3	Tesla Inc	TSLA	2.273108	1004.01	-4.77
4	Alphabet Inc. Class A	GOOGL	2.020859	2388.60	-107.69



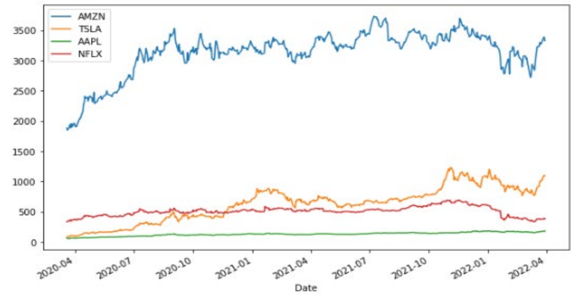
S&P Historical Data

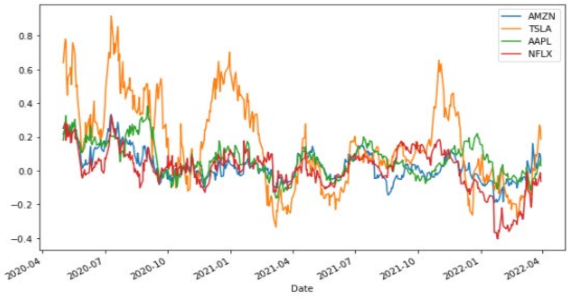
	AMZN	TSLA	AAPL	NFLX
Date				
2022-03-24	3272.989990	1013.919983	174.070007	375.709991
2022-03-25	3295.469971	1010.640015	174.720001	373.850006
2022-03-28	3379.810059	1091.839966	175.600006	378.510010
2022-03-29	3386.300049	1099.569946	178.960007	391.820007
2022-03-30	3326.020020	1093.989990	177.770004	381.470001

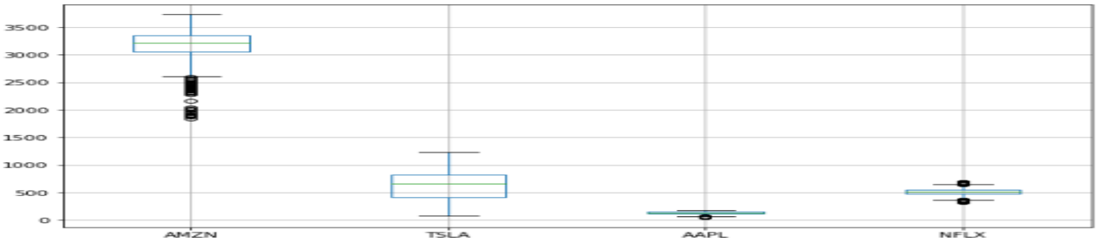
Methodology

Data Analysis

I started by analyzing stock market data, cleaning as needed and visualizing price and variability for select companies that are present in my dataset. Below are screenshots of visualizations. We can make some observations from visualizations, for example, we see that amazon has a lot more lower price drops compared to Apple or Netflix, may be because of higher stock price. Tesla stock has more volatility compared to Apple and it seems Netflix seems to be trending up after its lowest drop in two years..



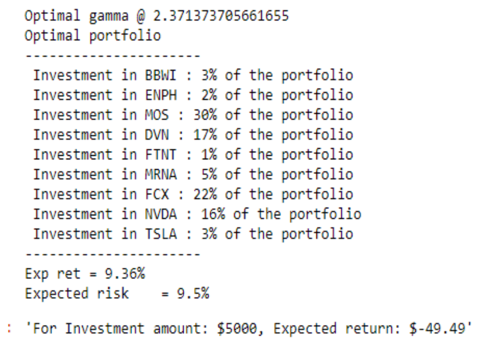




Portfolio Analysis

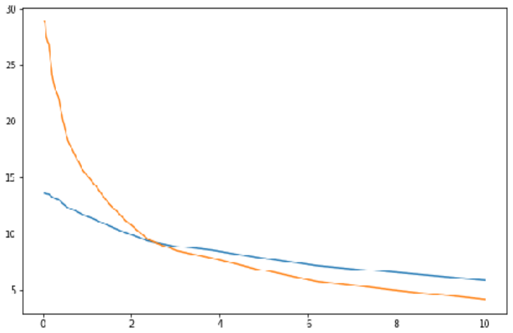
To run optimization, stock history dataset was converted from daily timestamp index to a business monthly start index, this was done to validate monthly gains, can be done for yearly as well. From this dataset, expected return ‘r’ and covariance ‘Q’ are calculated and objective function is run with many constrants constraints. using cvxpy package across entire S&P, returned best stocks by algorithm, screenshots attached below.

1) Results for simulation without any parameter with minimum return

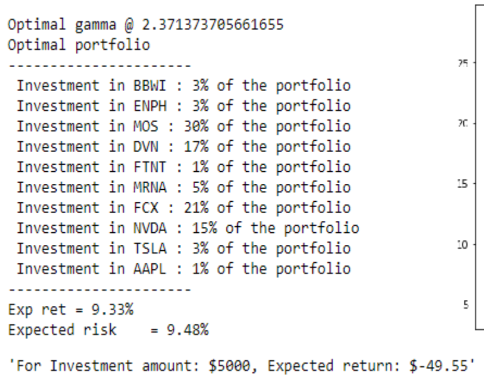


```
Optimal gamma @ 2.371373705661655
Optimal portfolio
---------------------
 Investment in BBWI : 3% of the portfolio
 Investment in ENPH : 2% of the portfolio
 Investment in MOS : 30% of the portfolio
 Investment in DVN : 17% of the portfolio
 Investment in FTNT : 1% of the portfolio
 Investment in MRNA : 5% of the portfolio
 Investment in FCX : 22% of the portfolio
 Investment in NVDA : 16% of the portfolio
 Investment in TSLA : 3% of the portfolio
---------------------
Exp ret = 9.36%
Expected risk    = 9.5%

: 'For Investment amount: $5000, Expected return: $-49.49'
```

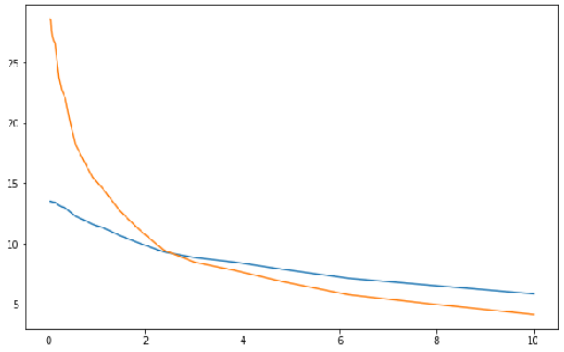


2) Results for simulation, select atleast one stock of Apple

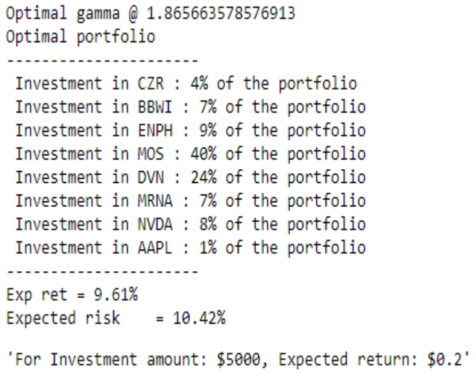


```
Optimal gamma @ 2.371373705661655
Optimal portfolio
---------------------
 Investment in BBWI : 3% of the portfolio
 Investment in ENPH : 3% of the portfolio
 Investment in MOS : 30% of the portfolio
 Investment in DVN : 17% of the portfolio
 Investment in FTNT : 1% of the portfolio
 Investment in MRNA : 5% of the portfolio
 Investment in FCX : 21% of the portfolio
 Investment in NVDA : 15% of the portfolio
 Investment in TSLA : 3% of the portfolio
 Investment in AAPL : 1% of the portfolio
---------------------
Exp ret = 9.33%
Expected risk    = 9.48%

'For Investment amount: $5000, Expected return: $-49.55'
```

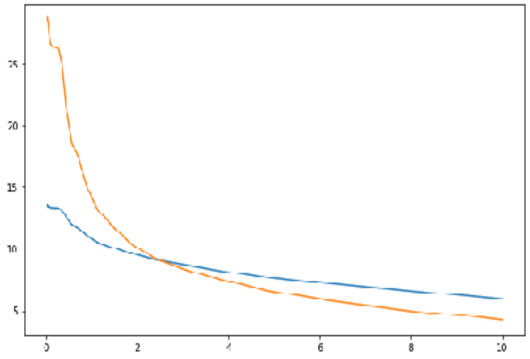


3) Results of simulation, select Apple, exclude Tesla, Freeport McMoRan



```
Optimal gamma @ 1.865663578576913
Optimal portfolio
---------------------
 Investment in CZR : 4% of the portfolio
 Investment in BBWI : 7% of the portfolio
 Investment in ENPH : 9% of the portfolio
 Investment in MOS : 40% of the portfolio
 Investment in DVN : 24% of the portfolio
 Investment in MRNA : 7% of the portfolio
 Investment in NVDA : 8% of the portfolio
 Investment in AAPL : 1% of the portfolio
---------------------
Exp ret = 9.61%
Expected risk    = 10.42%

'For Investment amount: $5000, Expected return: $0.2'
```



Single Stock Analysis

Ok, now that we have narrowed down stock selection to a few stocks by selecting trade off between returns and volatility, can we predict selected stock price? Can we remove highly risky stocks? Can we get any more insights on individual stocks?

For stock price prediction on selected stocks, I used Yahoo API to retrieve stock history data since 2015. This time data included Closing Price and Volume of stock per day.

Feature Engineering

Next, I created moving averages for 14, 30, 50 and 180 days then I created Relative Strength Index, i.e., average gain/average loss for 14, 30, 50 and 180 features from the closing price feature. Next, we create features from volume to capture interactions between 'Close Price' and 'Volume'. We will calculate moving averages from volume feature

New Features

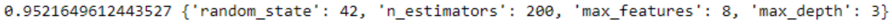
1) 10d_close_pct from close price,
2) ma14 from close price,
3) rsi14 from close price,
4) ma30 from close price ',
5) rsi30 from close price,
6) ma50 from close price,
7) rsi50 from close price,
8) ma180 from close price,
9) rsi180 from close price,
10) Adj_Volume_10d_change,
11) Adj_Volume_10d_change_SMA

Predictive Analytics

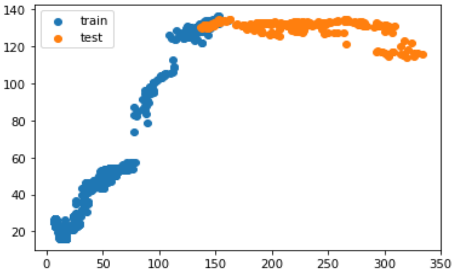
To run predictive analysis, I divided my data into train, test splits. After lengthy research, literature pointed to various pitfalls of choosing randomized data for timeseries data. It was recommended to split data at a certain point in time thereby keeping their sequence for testing. So, data was split at around 80-20 split.

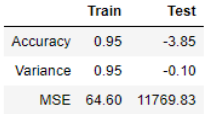
Random Forest Regression

Random Forest Regressor was used to train and perform prediction. After choosing the best parameters for the model, testing was performed on test set. Results were very disappointing. Training accuracy was around 95.68%, however prediction accuracy on test data was very low.

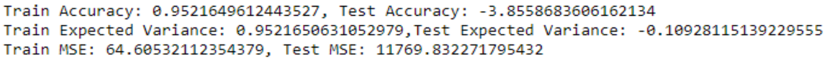


```
0.9521649612443527 {'random_state': 42, 'n_estimators': 200, 'max_features': 8, 'max_depth': 3}
```





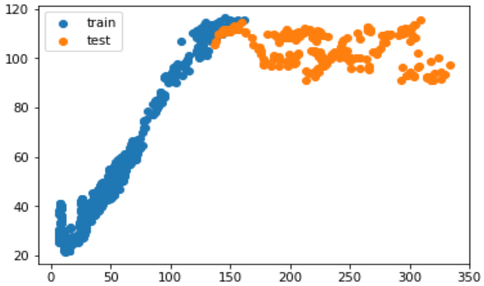
	Train	Test
Accuracy	0.95	-3.85
Variance	0.95	-0.10
MSE	64.60	11769.83

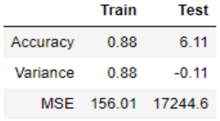


```
Train Accuracy: 0.9521649612443527, Test Accuracy: -3.8558683606162134
Train Expected Variance: 0.9521650631052979,Test Expected Variance: -0.10928115139229555
Train MSE: 64.60532112354379, Test MSE: 11769.832271795432
```

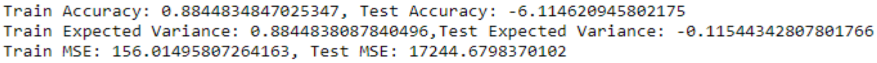
Gradient Boosting Regression

Gradient Boosting Regression was used to train and perform prediction. Results were very disappointing as well. Training accuracy was around 88.87%, however prediction accuracy on test data was very low.





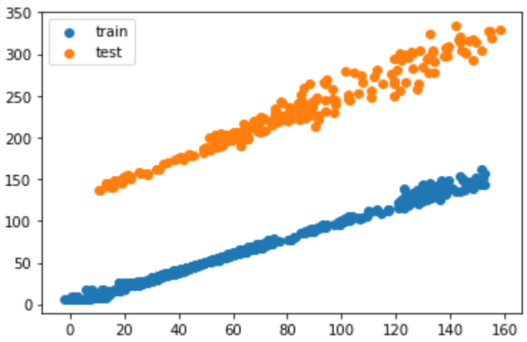
	Train	Test
Accuracy	0.88	6.11
Variance	0.88	-0.11
MSE	156.01	17244.6

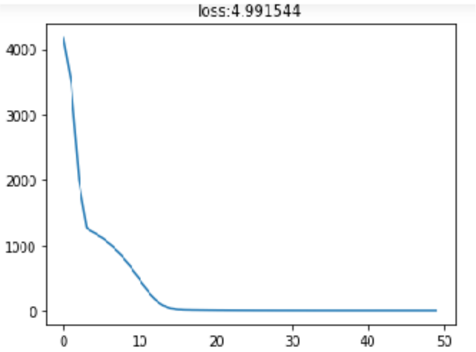


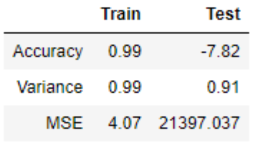
```
Train Accuracy: 0.8844834847025347, Test Accuracy: -6.114620945802175
Train Expected Variance: 0.8844838087840496,Test Expected Variance: -0.11544342807801766
Train MSE: 156.01495807264163, Test MSE: 17244.6798370102
```

Neural Network w/Custom Loss Function

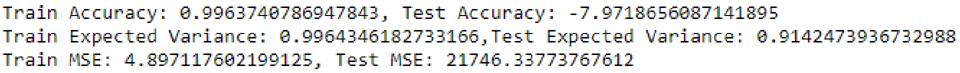
Neural network was used to train and perform prediction and I had the same results as the above non-linear models. Accuracy around 99% however failed to predict test data.







	Train	Test
Accuracy	0.99	-7.82
Variance	0.99	0.91
MSE	4.07	21397.037



```
Train Accuracy: 0.9963740786947843, Test Accuracy: -7.9718656087141895
Train Expected Variance: 0.9964346182733166,Test Expected Variance: 0.9142473936732988
Train MSE: 4.897117602199125, Test MSE: 21746.33773767612
```

Evaluation:

For portfolio optimization problem once stocks are selected, we use some arbitrary amount for investment (in this case $5000), weights of investment from optimization, % change for the last month in data set, are used to calculate returns. i.e. R = Investment – (w1*stock1 + w2*stock2…… wn*stockn). For various simulation there were negative return and some simulation positive returns.

For Machine learning classifiers, Random Forest, Gradient Boost and Neural Network about 15% of single stock dataset was used to evaluate results on engineered features. All the models had bad performance predicting future of a single stock price.

Conclusion:

We used **real** stock market historical data over 500 companies to build project portfolio and run optimization. We then split our data and performed predictive analysis for hoping to get some insight into individual stocks. We evaluated returns by validating investment on last month's return. For most the project I felt like having a parallel processing would help with calculations to train models. I started using LSTM model for stock price prediction and very soon it became clear, that it is **impossible** to **predict** stock prices or select best portfolio based on previous price or data alone. Even though we used price and volume to engineer some features and run predictive analysis. This alone is not enough, and for **next** steps, we need to incorporate a lot more features based on sentiment analysis, geographic relations between countries, future events like pandemic, supply chain issues etc. to build better predictive models.

Maybe it would help making friends with policy makers :D