PORTFOLIO ANALYSIS AND OPTIMIZATION: A DATA-DRIVEN APPROACH

Ngoc Tran

DSA 5303 Financial Engineering Final Project

INTRODUCTION

- Modern Portfolio Theory – Harry Markowitz (1952)
  - Efficient Frontier
  - Sharpe Ratio
  - Diversification
- Monte-Carlo simulation
- Future stock price diagnostic

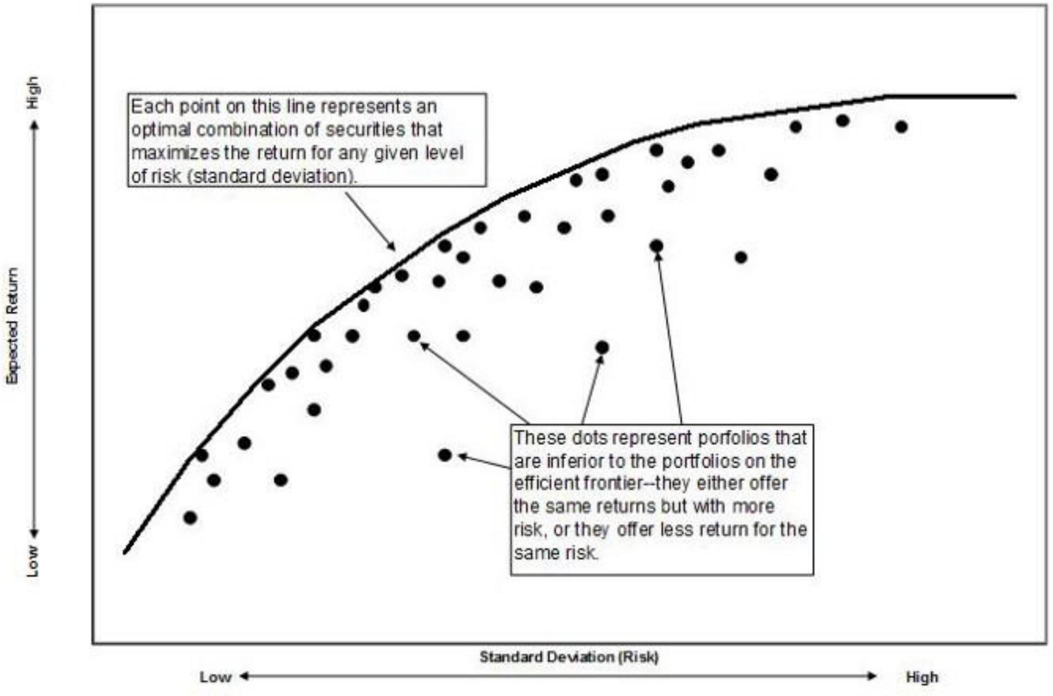
MODERN PORTFOLIO THEORY (MPT)

HIGH RISK
HIGH REWARD

- Assumptions:
  - Investors are rational and avoid risk when possible
  - Investors aim for maximum return for their investment
  - All investors share the aim maximizing expected return
  - Commission and taxes are not put into consideration
  - All investors have access to same sources and level of all necessary information about decision
  - Investors have unlimited access to borrow and lend at risk-free rate

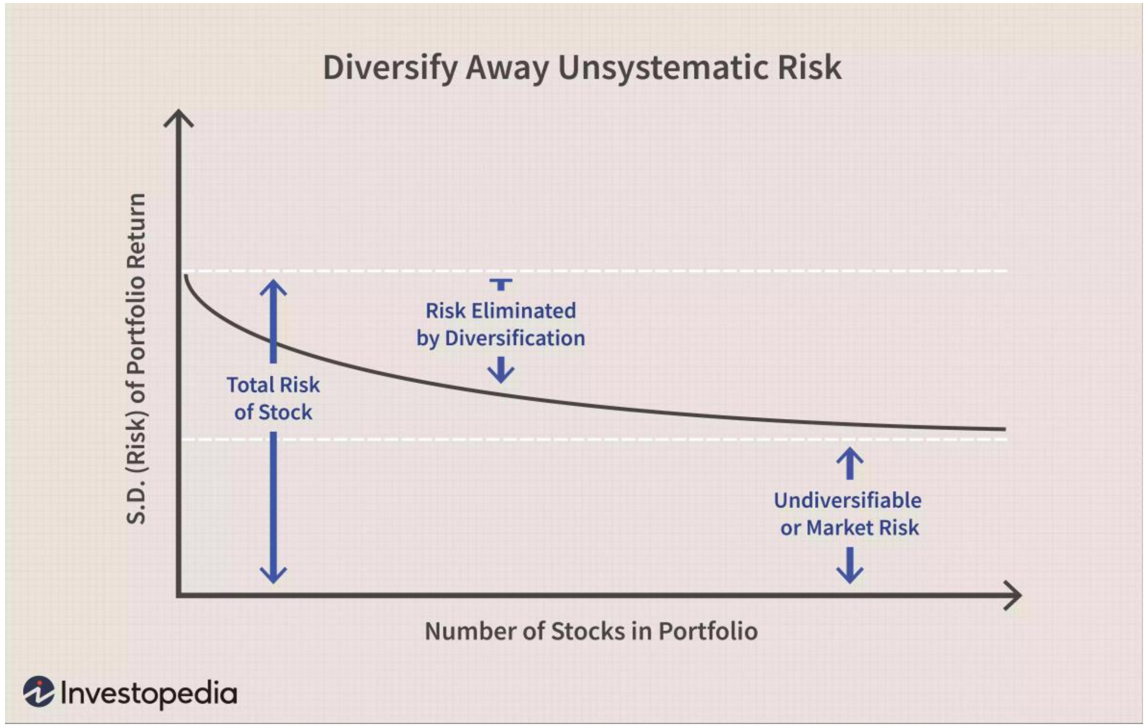
EFFICIENT FRONTIER AND SHARPE RATIO

- Efficient set of portfolios
  - Highest expected return for certain risk
  - Lowest risk for a given expected return
- Adjust portfolio's past and expected future performances
  - High Sharpe ratio is good when compared to similar portfolios with lower returns
  - Normally distributed investment return



DIVERSIFICATION

- MPT advocates for diversification of securities and assets
- MPT extends the diversification concept that differentiate non-systematic risk (i.e. Proper diversification can dampen unsystematic risk)

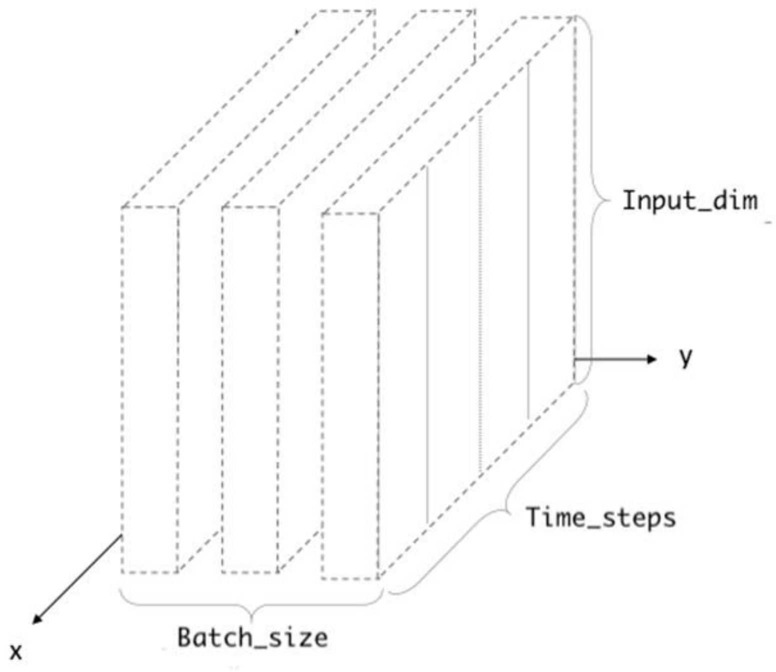


MONTE-CARLO SIMULATIONS

- Most common approach for portfolio management
- Carrying out repeated trials by using randomly generated inputs (i.e. conducting numbers of iterations by using random weights) and observing the outcomes from complex scenarios based on probabilities of certain events occurring
- Higher iterations, higher optimization accuracy
- Use simulation to figure out allocation for each stock and visualize the returns of each stocks against volatility (risk) to establish shape of EF
- 10000 iterations ~ stock scenarios ~ numbers of stocks

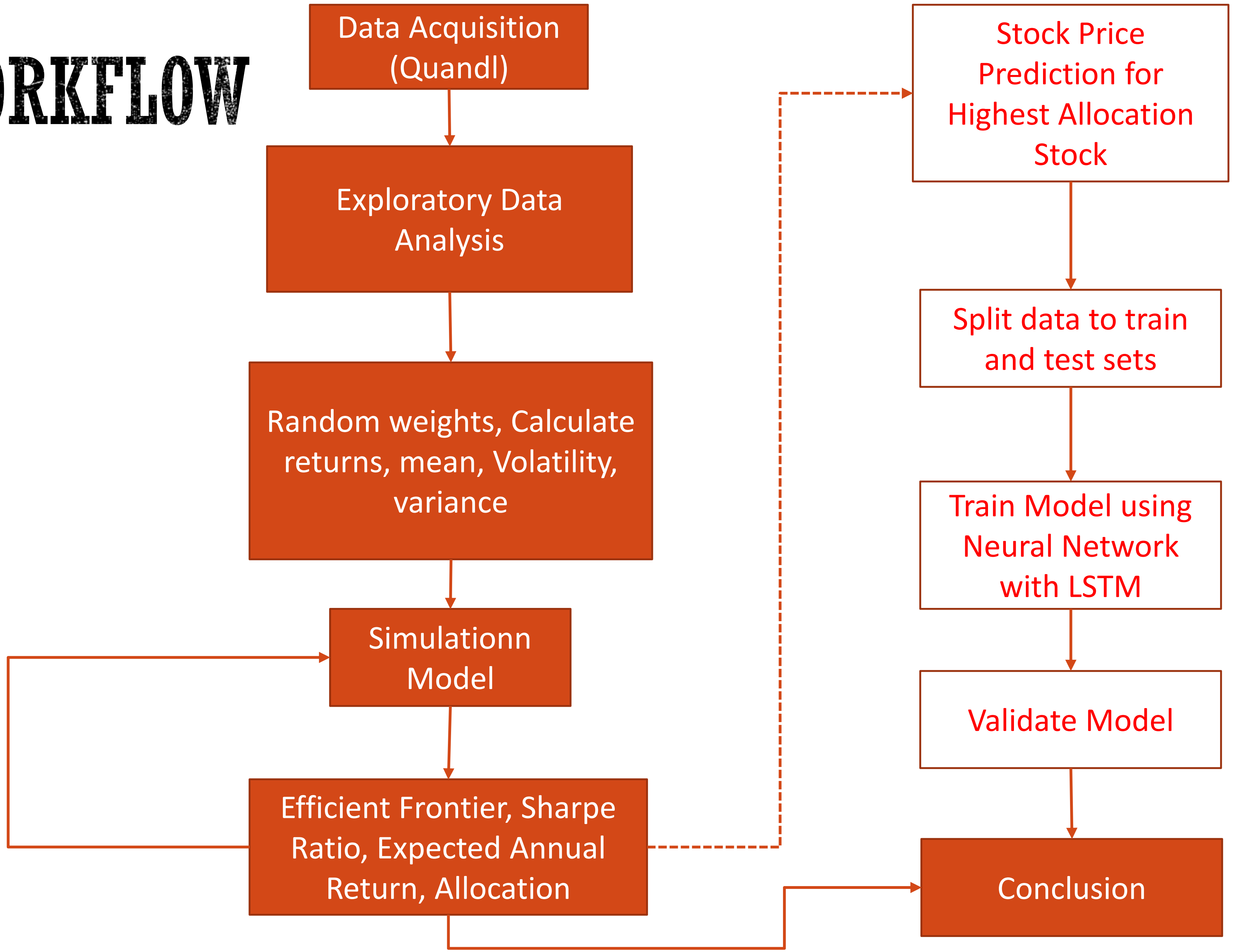
LONG-TERM SHORT TERM MEMORY (LSTM) FOR PRICE PREDICTION

- Sequence prediction problems due to past storage capability
- 3-dimensional input data structures:
  - Batch-size
  - Time-steps
  - Numbers of features in one input sequence
- Model architectures in Keras:
  - Sequential -> invoke neural network
  - LSTM
  - Dropout rate



Input shape for LSTM network

WORKFLOW


Data Acquisition (Quandl)
Exploratory Data Analysis
Random weights, Calculate returns, mean, Volatility, variance
Simulationn Model
Efficient Frontier, Sharpe Ratio, Expected Annual Return, Allocation
Stock Price Prediction for Highest Allocation Stock
Split data to train and test sets
Train Model using Neural Network with LSTM
Validate Model
Conclusion

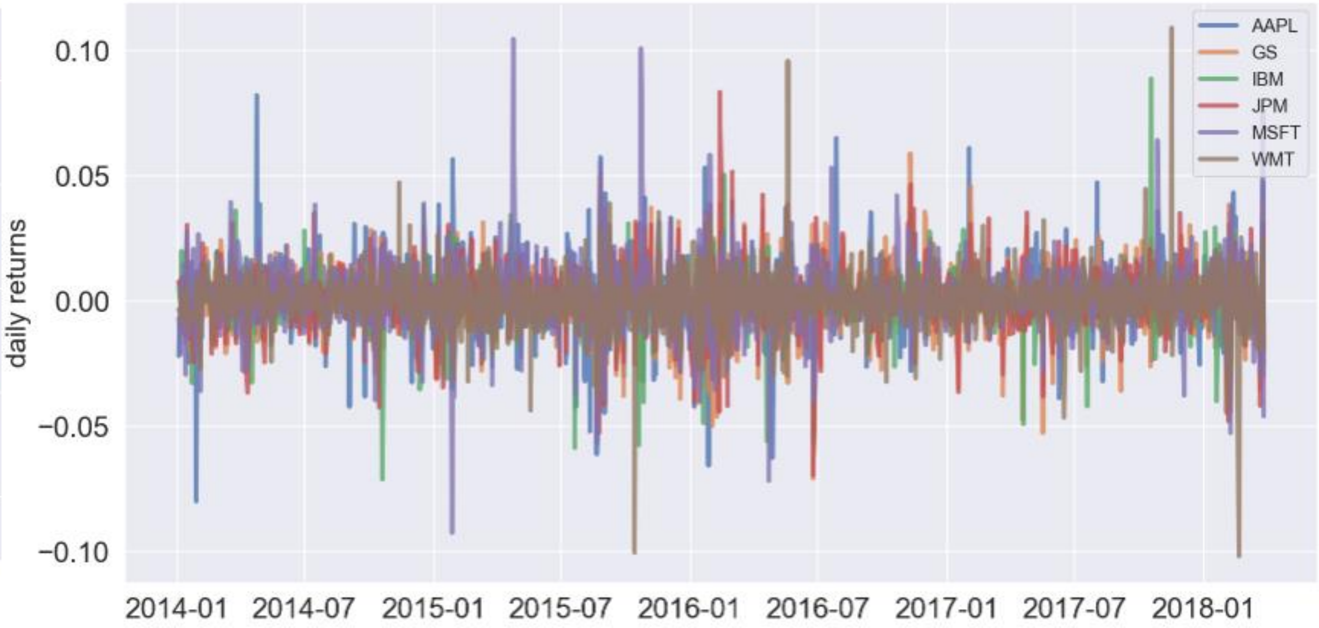
EXPLORATORY DATA ANALYSIS

- Start date: Jan 2014
- Stocks: Apple (APPL), IBM, Microsoft (MSFT), Goldman Sachs (GS), JP Morgan Chase (JPM), Walmart (WMT)
- Adjusted close date

date	AAPL	GS	IBM	JPM	MSFT	WMT
2014-01-02	73.523423	168.112371	162.670896	52.761509	33.532800	71.343743
2014-01-03	71.908415	169.309847	163.644133	53.169389	33.307202	71.108673
2014-01-06	72.300536	170.469309	163.082988	53.477565	32.603338	70.710863
2014-01-07	71.783135	169.442900	166.335879	52.861213	32.856007	70.927850
2014-01-08	72.238063	169.585457	164.810264	53.359733	32.269454	70.367299

EXPLORATORY DATA ANALYSIS

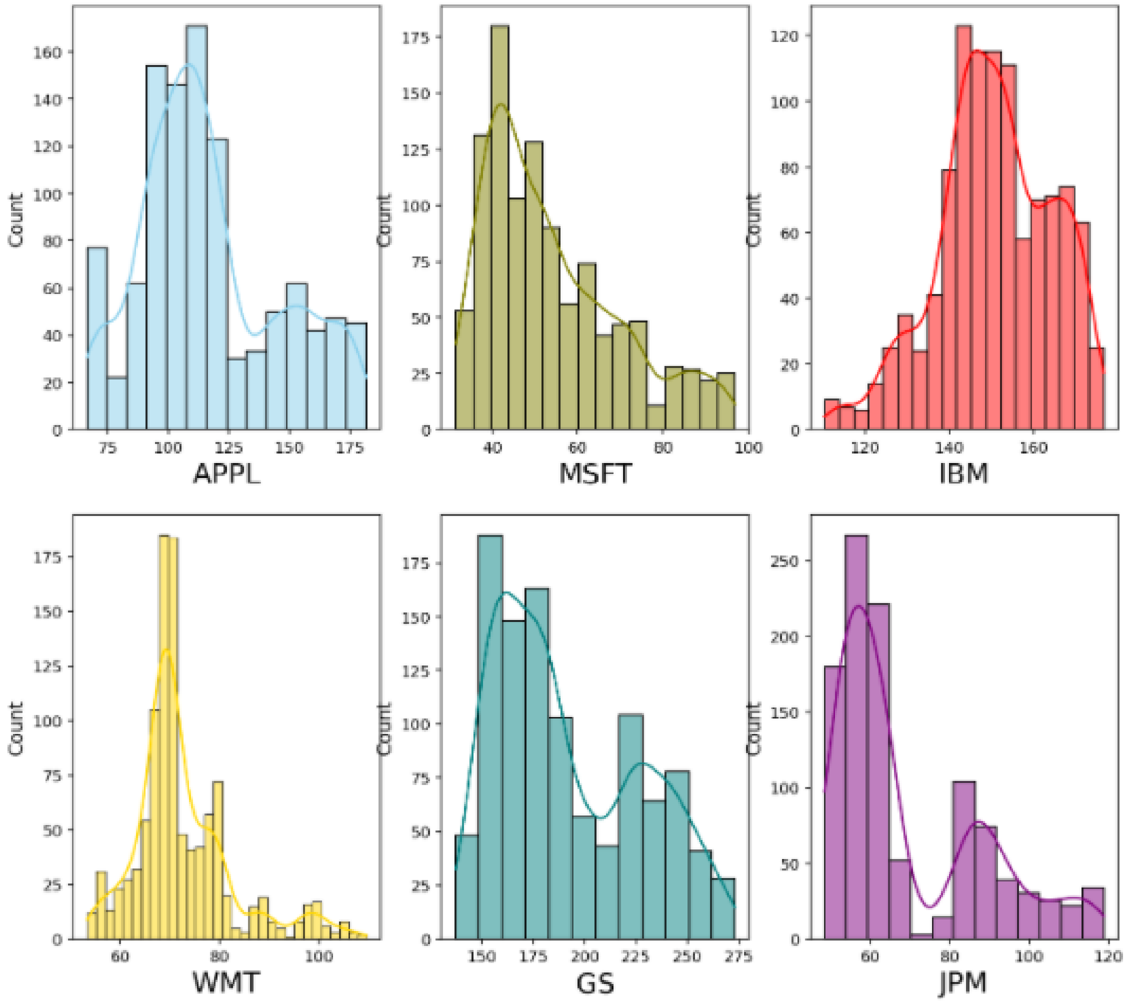
Prices of Stock

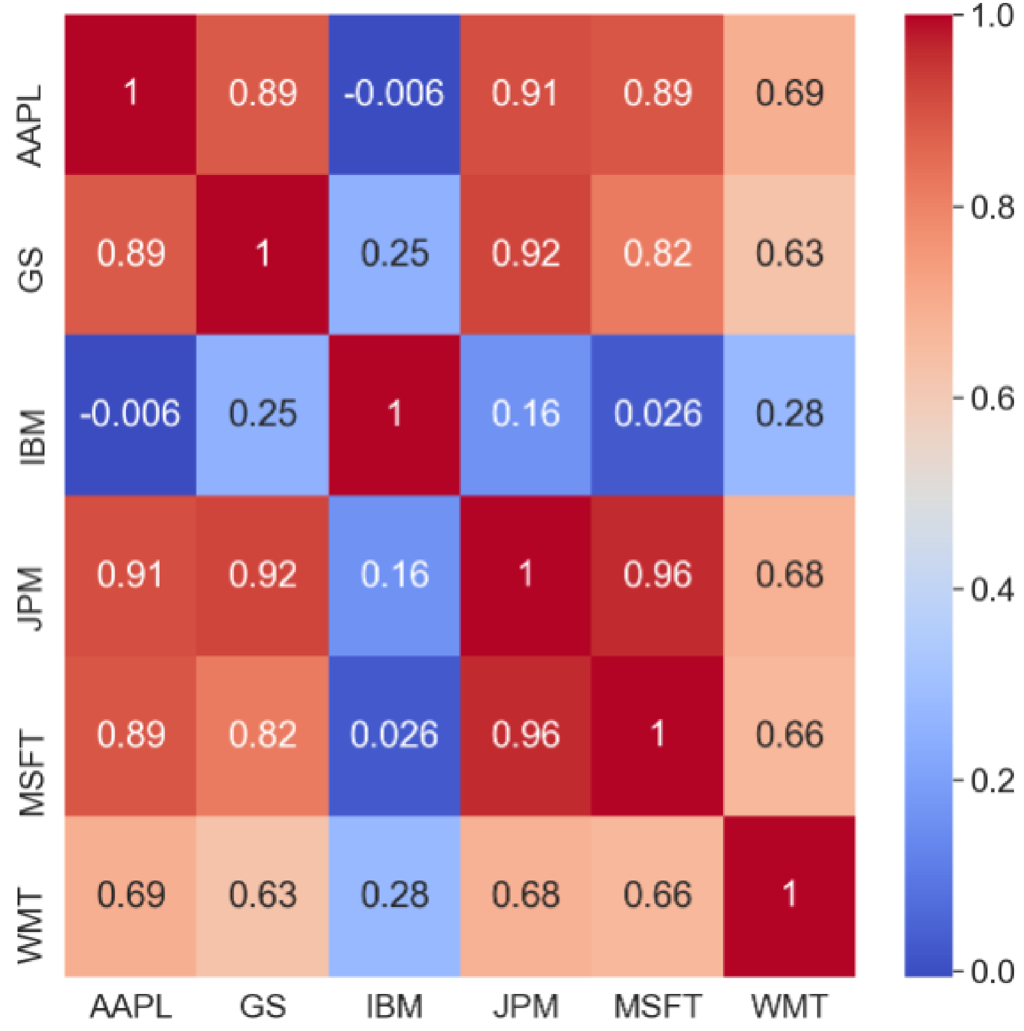


Daily Return

$$\text{Daily return} = \frac{new\ \ price - old\ price}{old\ price}$$

EXPLORATORY DATA ANALYSIS

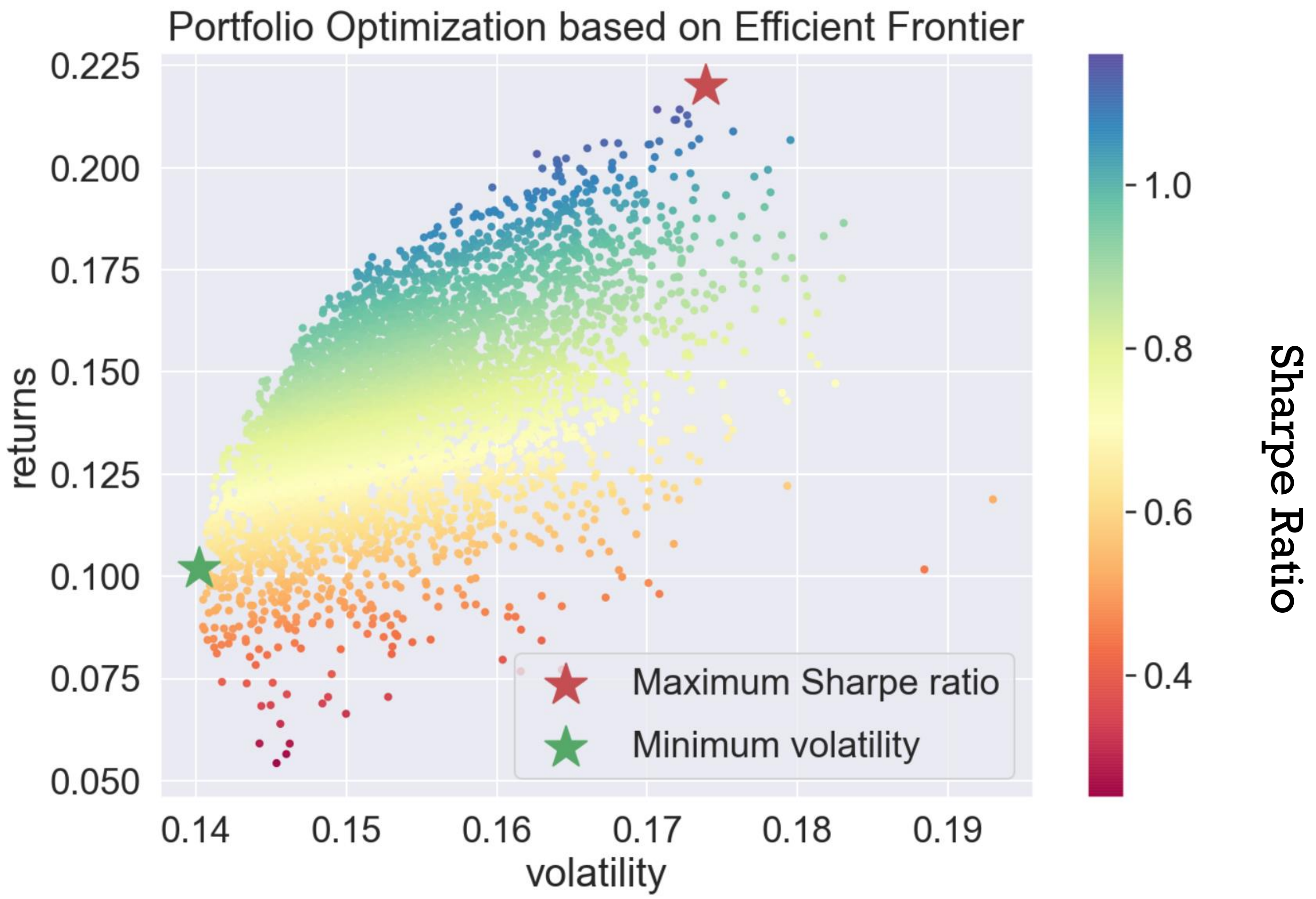




INPUTS AND FORMULAS

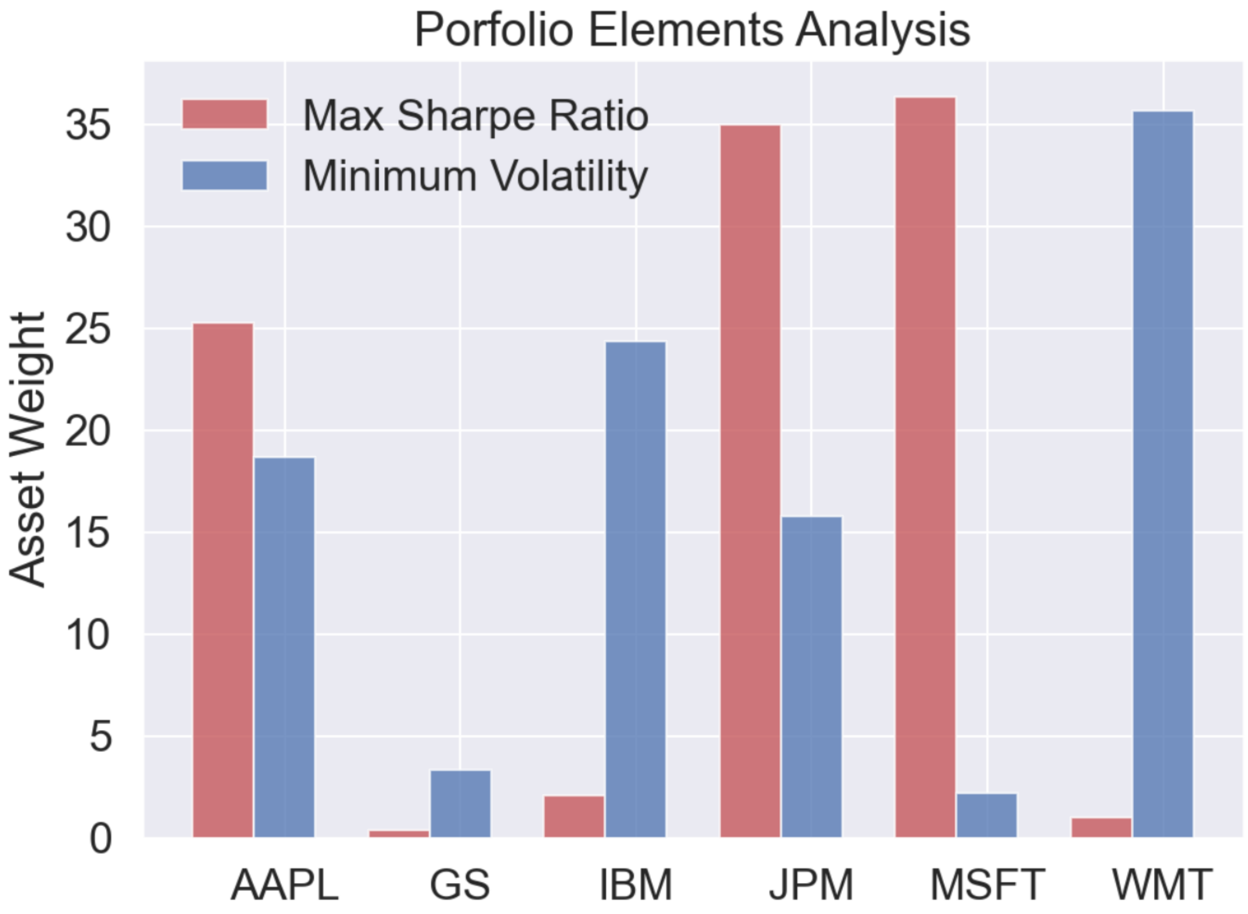
- $Total\ Weights = w_1 + w_2 + w_3 + .. + w_6 = 1$ -> constraints
- $Mean(portfolio\ return) = \sum_{i=1}^{n} w_i r_i$
- $Variance(porfolio\ return) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \rho_{i,j} \sigma_i \sigma_j = w^T \times (cov \times w)$
- $Volatility = \sqrt{w^T \times (cov \times w)}$
- $Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$
  - (Rp: return of portfolio, Rf: risk-free rate, σ_P: s.t.d of portfolio's excess return)
- Scipy's optimize function, method SLQP -> minimize negative Sharpe ratio - $\frac{R_p - R_f}{\sigma_p}$

RESULTS: INITIAL RANDOMLY SIMULATION



	Maximum Sharpe Ratio Portfolio Allocation	Minimum Volatility Portfolio Allocation
Return	0.22	0.1
Volatility	0.18	0.14

RESULTS: INITIAL RANDOMLY SIMULATION



Maximum Sharpe Ratio Portfolio Allocation

	AAPL	GS	IBM	JPM	MSFT	WMT
allocation	18.46	1.75	0.14	31.79	45.85	2.01

Minimum Volatility Portfolio Allocation

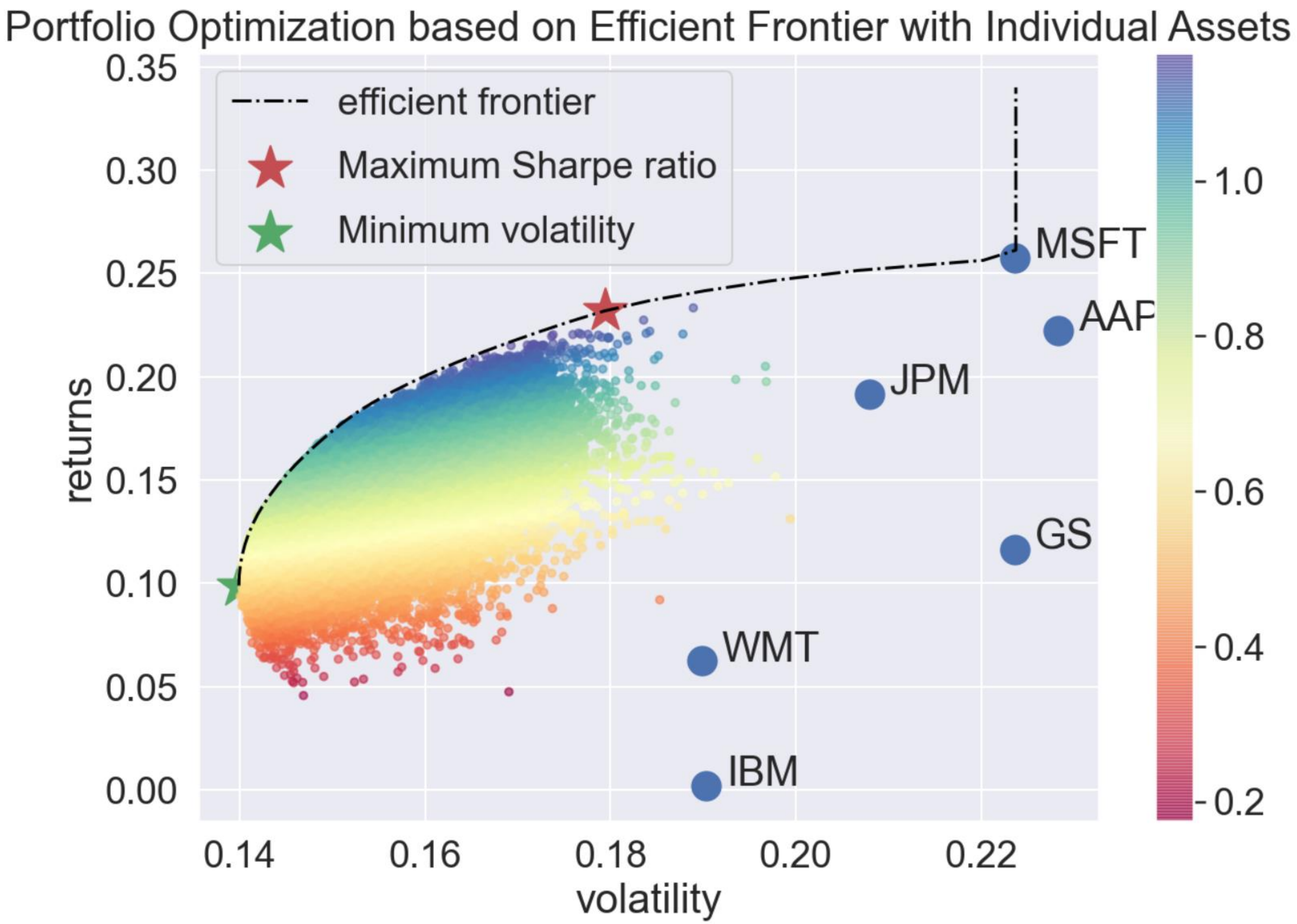
	AAPL	GS	IBM	JPM	MSFT	WMT
allocation	14.66	6.72	26.74	8.45	8.67	34.75

RESULTS: OPTIMIZED SIMULATION

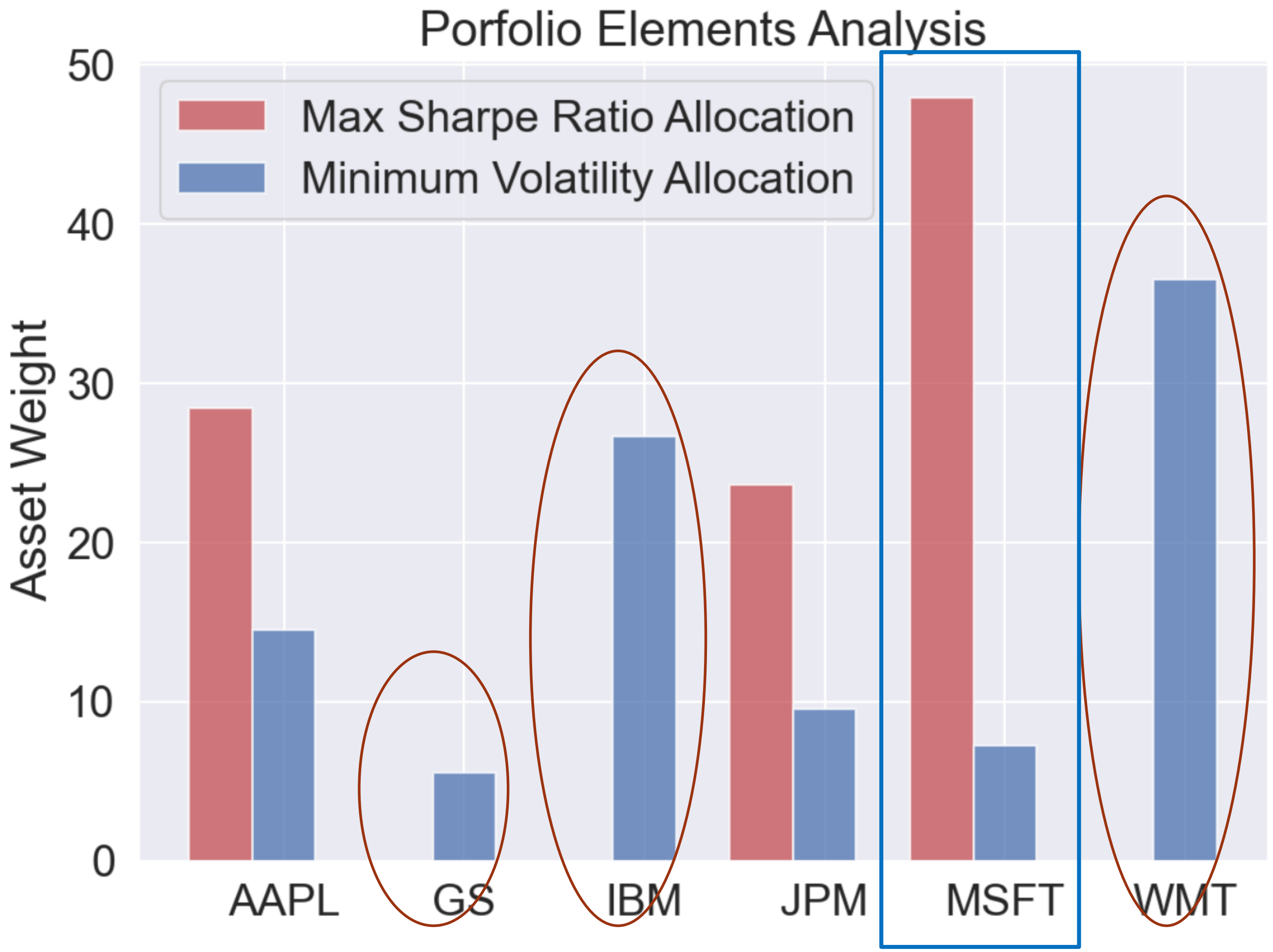
Stock	APPL	GS	IBM	JPM	MSFT	WMT
Risk	0.23	0.22	0.19	0.21	0.22	0.19

	Maximum Sharpe Ratio Portfolio Allocation	Minimum Volatility Portfolio Allocation
Return	0.24	0.1
Volatility	0.18	0.14

- Max Sharpe Ratio: 1.2



RESULTS : OPTIMIZED SIMULATION

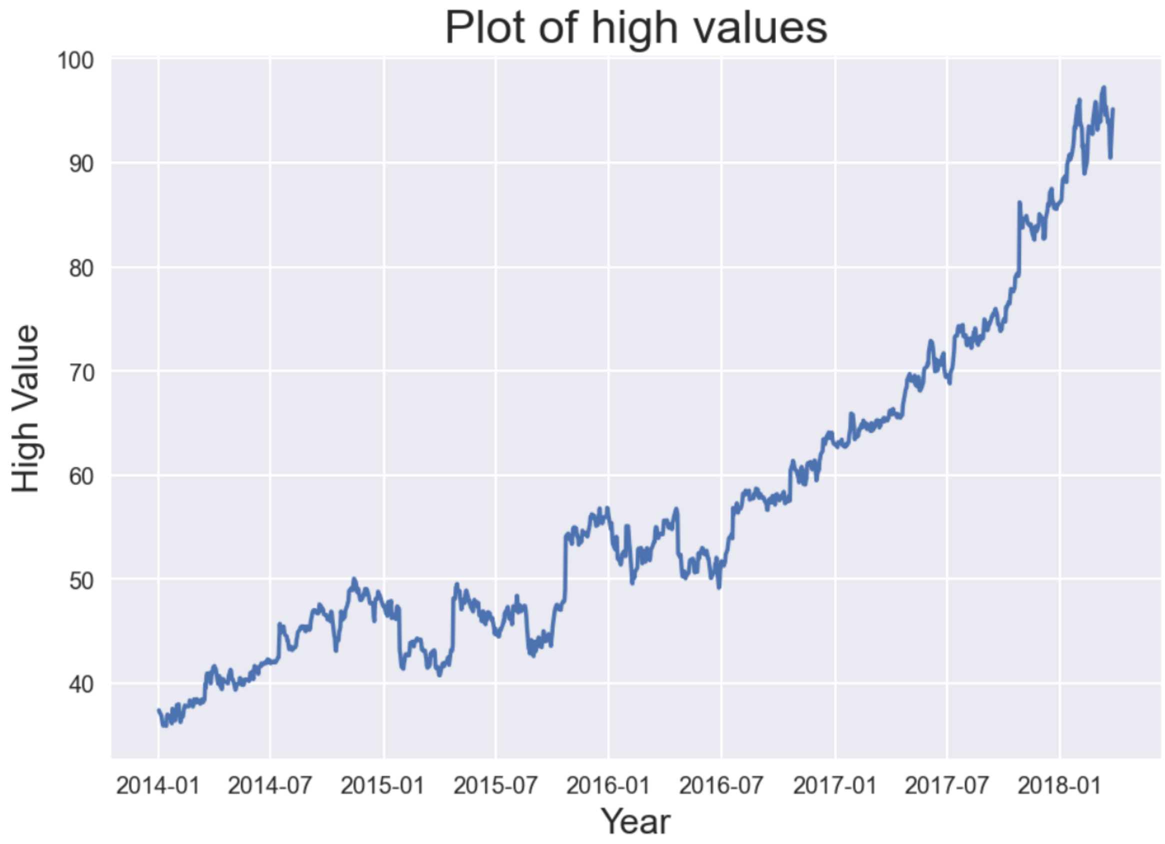


Maximum Sharpe Ratio Portfolio Allocation

	AAPL	GS	IBM	JPM	MSFT	WMT
allocation	28.425	0.0	0.0	23.63	47.945	0.0

- GS: high volatility and less return
- WMT and IBM: low covariance, low risk and less return, low value
- MSFT and APPL: higher risk, high return
- JPM: less risk than GS but higher return

LSTM APPLICATION ON HIGH-VALUE STOCK PREDICTION



- Inputs:
  - Using StandardScaler()
- The high value of current day depends on the trends in last 60 days
  - timestep = 60
  - X index: i to (i+timestep)
  - Y index: i+timestep
- Train/ Test split 65%
  - Convert to 3D tensor

```
print(X_train.shape)
print(x_test.shape)
```

```
(631, 60, 1)
(312, 60, 1)
```

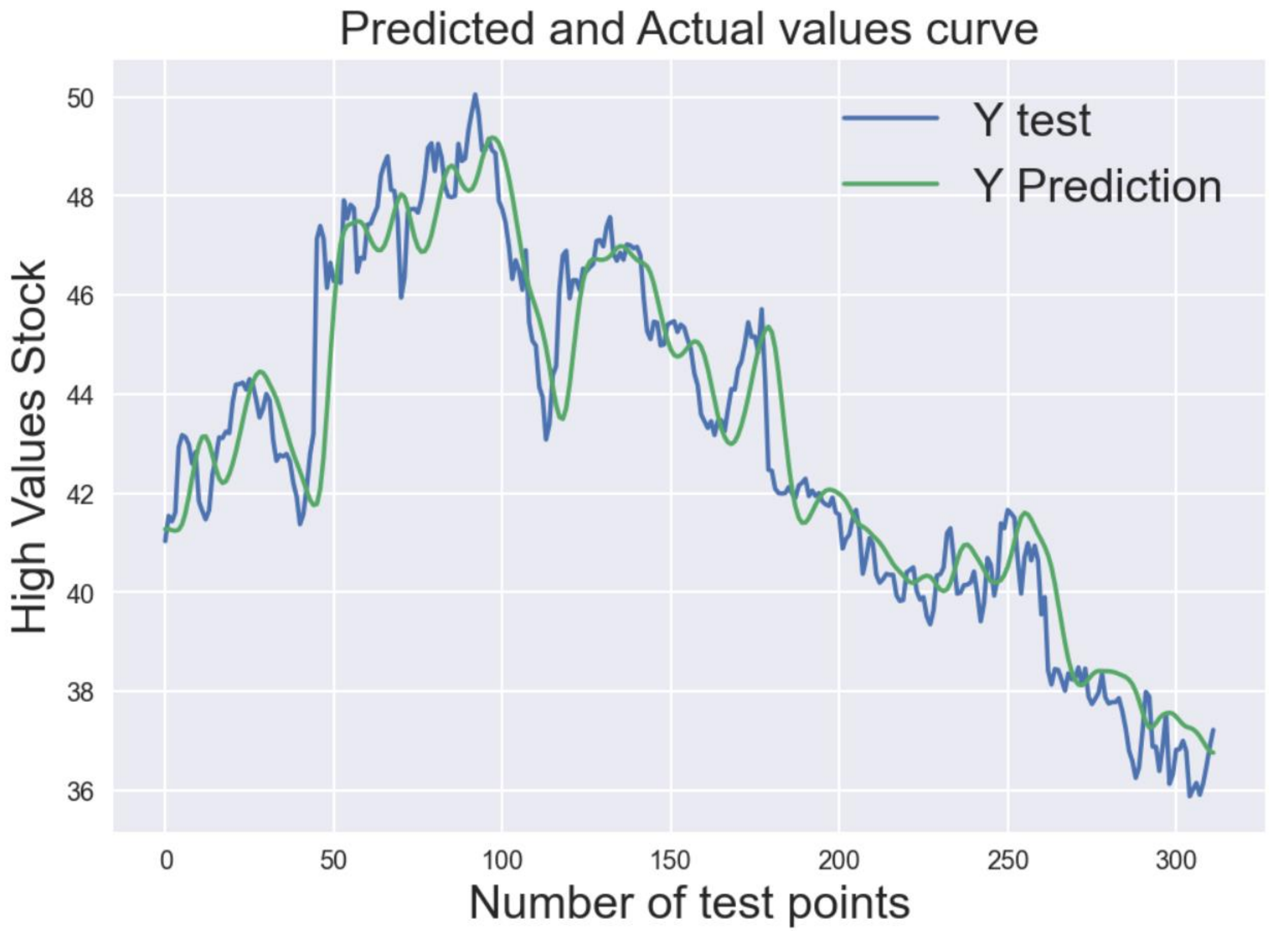
LSTM APPLICATION ON HIGH-VALUE STOCK PREDICTION

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 60, 64)            16896
_________________________________________________________________
lstm_1 (LSTM)                (None, 60, 50)            23000
_________________________________________________________________
lstm_2 (LSTM)                (None, 60, 32)            10624
_________________________________________________________________
lstm_3 (LSTM)                (None, 50)                16600
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 67,171
Trainable params: 67,171
Non-trainable params: 0
_________________________________________________________________
```

```python
from sklearn.metrics import mean_squared_error

loss = np.sqrt(mean_squared_error(y_test,y_pred))
loss
```

```
0.07008533871771033
```



CONCLUSIONS

- Using MonteCarlo simulation, the optimal portfolio clarifies the efficient of diversification and confirms the improvement in the returns by having riskier stocks
- The portfolio allocation in this project allows the investors can predict the return of individual stocks that they invest
- The analysis of future stock price using LSTM is helpful to understand the impeding trend in future with the loss of 0.07
- Comprehensive studies in stock performance diagnostic using ML techniques

LINK TO PRESENTATION RECORD

- https://mymedia.ou.edu/media/DSA+5303+Final+Project+%28Updated%29+Ngoc+Tran/1_sab5q44d

REFERENCES

- https://shiva-verma.medium.com/understanding-input-and-output-shape-in-lstm-keras-c501ee95c65e
- https://medium.com/python-data/effient-frontier-in-python-34b0c3043314