How feasible can the Support Vector Machine algorithm and the Random Forest algorithm be in the prediction of future stock prices of companies listed on the S&P 500?

Implementing Machine Learning algorithms in the stock market

Jonathan Vincentius

Table of Contents

1 Introduction

Machine Learning is an application of Artificial Intelligence that provides computer systems the ability to automatically learn and improve from experience in performing a desired action, without explicit programming or human intervention. In other words, they form correlations and detect patterns in historical datasets to become more accurate at predicting future output values.

Due to the world's ever-evolving technology and generation of huge amounts of data, its need is now at an all time high, and will only continue to grow in the future. Machine learning has paved the way for technological accomplishments that would have been impossible a few years ago, it is the underlying technology behind image recognition, self-driving cars, virtual assistants, and is already being used in a wide range of applications in countless industries, especially Finance: risk level evaluation, credit score calculation, and fraud detection all involve processing enormous volumes of data relating to daily transactions, bills, vendors, and customers (Butcher, 2017). Machine learning algorithms are more accurate in drawing insights and making predictions when larger volumes of training data are input since they provide more accurate mean values, identify outliers, and provide a smaller margin of error. This all makes it increasingly feasible for use in the Financial markets, becoming a key aspect of numerous financial services and applications. (CFI, 2021)

This paper seeks to investigate its feasibility in one application of finance specifically – stock trading. This practise is known as algorithmic trading in the investing world, and it involves using pre-programmed instructions to analyse various parameters while simultaneously monitoring business news in real-time to detect factors that may fluctuate security prices, to make trade decisions on financial securities to generate returns.

In the past few years, I've personally had experience with the struggles of attempting to trade stocks and come out profitable. A study (Jordan & Diltz, 2019) statistically proved that the number of day traders losing money is twice that of those that make money. Reasons may include poor trading strategy, lack of risk management, emotional influence, etc. All these limitations are eliminated in algorithmic trading, with the upside that it can analyze larger volumes of stock data, deduce lightning fast trade decisions, and execute thousands of trades in a single day.

There is a lot of uncertainty and debate around the ability of machine learning to consistently outperform humans in generating profits. There remains the question of how these algorithms will cope should the market analysis required surpass the analytical capabilities programmed. What is their true potential in actually generating accurate predictions? Will its computational complexities hinder it from becoming feasible in the real markets? Is there even a remote possibility of creating a sophisticated enough machine learning trading system that can consistently outperform the market, and potentially break it?

In an attempt to resolve this uncertainty, this paper will investigate the feasibilities of 2 machine learning algorithms in predicting future stock prices. This paper will compile programs to extensively test each algorithm over a sufficiently large and diverse dataset, before evaluating their prediction performances and other characteristics to determine their feasibility for stock trading. Stock markets play a principal role in economic development and improving our standards of living. Raising capital allows companies to expand operations, provide better goods and services, create jobs, and ultimately maximise society’s well being.

This paper hopes to grant further insight into machine learning and open up possibilities of utilising it to generate better returns in the financial markets for all, whether it's for multinational corporations, institutional investors, or even ordinary retail investors.

2 Theoretical Background

2.1 Machine Learning and Algorithmic Trading

Machine Learning algorithms are categorized by how it learns to become more accurate in its predictions, and the type of data one wishes to predict. The two main approaches being supervised learning and unsupervised learning. (Brownlee, 2019)

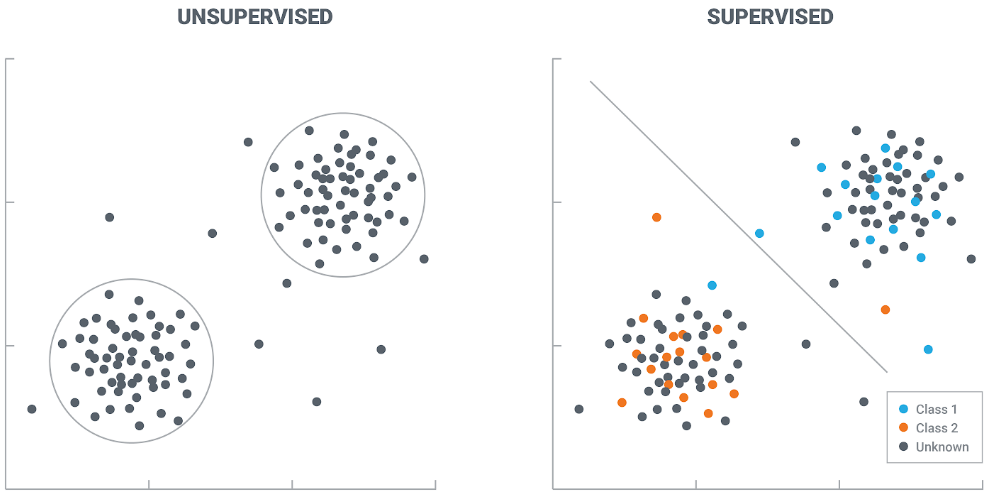

Figure 1: Visualising supervised & unsupervised learning

"Supervised Learning vs Unsupervised Learning. Which Is Better?" Info@lawtomated.com. April 8 2019

<u>Supervised Learning</u>

Examples: [Random Forest, Support Vector Machine, Nearest Neighbour]

Algorithms are supplied with labelled training data and given defined variables for which they will assess for correlations. It repeatedly makes predictions on the training set and compares them to the known output values, to optimize this mapping function and obtain the best predictions. Stopping only when an appropriate level of success is achieved. In Finance, algorithms are provided with historical data and tasked to find the relationship with the best predictive power. Supervised learning can be split into two sets of problems, classification and regression. Classification has categorical (qualitative) output variables, regression has numerical outputs (quantitative) (Burns, 2020) (Bushkovskyi, 2018). Stock price prediction can fall under either one, predicting general rise or fall would be classification and predicting the numerical price would be regression. Regression will be utilised in the investigation because in a practical sense, it is much more beneficial to be able to predict the numerical change in price. Realistically, predicting general price direction change is futile knowing the exact change.

Unsupervised Learning

Examples: [K-means Clustering, t-SNE, PCA, Association Rule]

Algorithms are trained on unlabeled data. They scan through datasets looking for meaningful patterns and underlying structures. There are no true correct answers to map the model to, so the algorithm creates its own methods within the training set. In Finance, they are supplied with sets of values from assets, to identify the potential drivers of the data (momentum/volatility/liquidity).

This paper will be investigating the Support Vector Machine and Random Forest algorithms. Besides being among the most commonly used algorithms, a study (Lu, Yuan, Li & Xiang, 2019) has also indicated that these two yielded more accurate predictions than the likes of naive bayes and logistic regression. Selecting the best performing algorithms allow us to better assess the maximum predictive potential of machine learning as a whole, making them suitable as representatives. (Venables, 2020)

Aspects of an Algorithmic Trading System

When developing a trading system, the following elements must be determined: (Seth, 2020) Target markets, Trading Logic, and parameters by which the trading logic will be triggered. Ideally, the system should account for the cost of trading and commission, but since they depend on the trading platform, for simplicity's sake it will be disregarded in this investigation. The target market will be limited to the S&P 500 index for reasons that will be divulged upon in an ensuing section. Trading logic will be the 2 aforementioned algorithms. Input parameters will be historical adjusted close stock prices. Ideally, a more sophisticated system should also take in other non-price input parameters, such as recent news around the company, market sentiment, recent balance sheets/financial statements, etc. Investor confidence has a large impact in fluctuating the long term trends of security prices, and the company's financial information is an important key factor that all adept investors analyse carefully before taking positions in any company's stock. This will not be implemented due to lack of expertise, and for simplicity's sake.

Once any Algorithmic Trading System has been fully developed, it must be back tested before implementation in the live markets, in other words, testing it on historical periods of past stock-market performance to see if using it would have been profitable. This is a crucial step for anyone programming a system of their own, and essentially what will be done for the investigation.

The diagram below provides a quick summary of all the previously mentioned points and an outline on the basic functionalities of an Algorithmic Trading System.

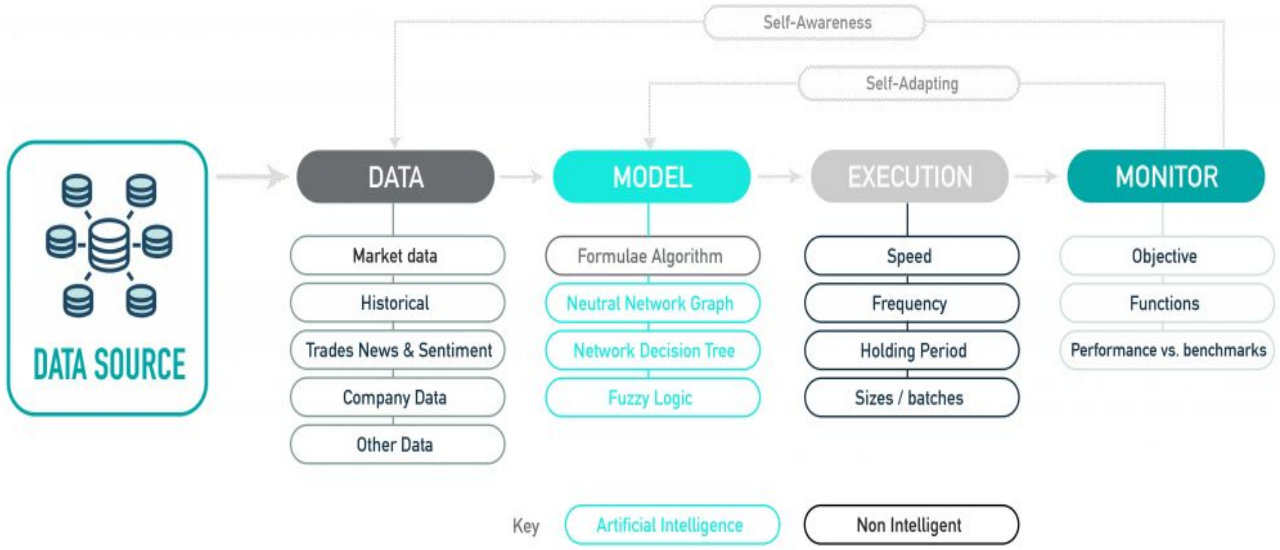

Figure 2: Conceptual Model of Algorithmic Trading "AI." Brown Armadillo 2012

2.2 Support Vector Machine Algorithm (SVM)

The Support Vector Machine algorithm is a supervised, linear modelled algorithm utilised mostly for classification, but can also be used for regression. The algorithm plots the data points and outputs the hyperplane or a boundary that best separates them.

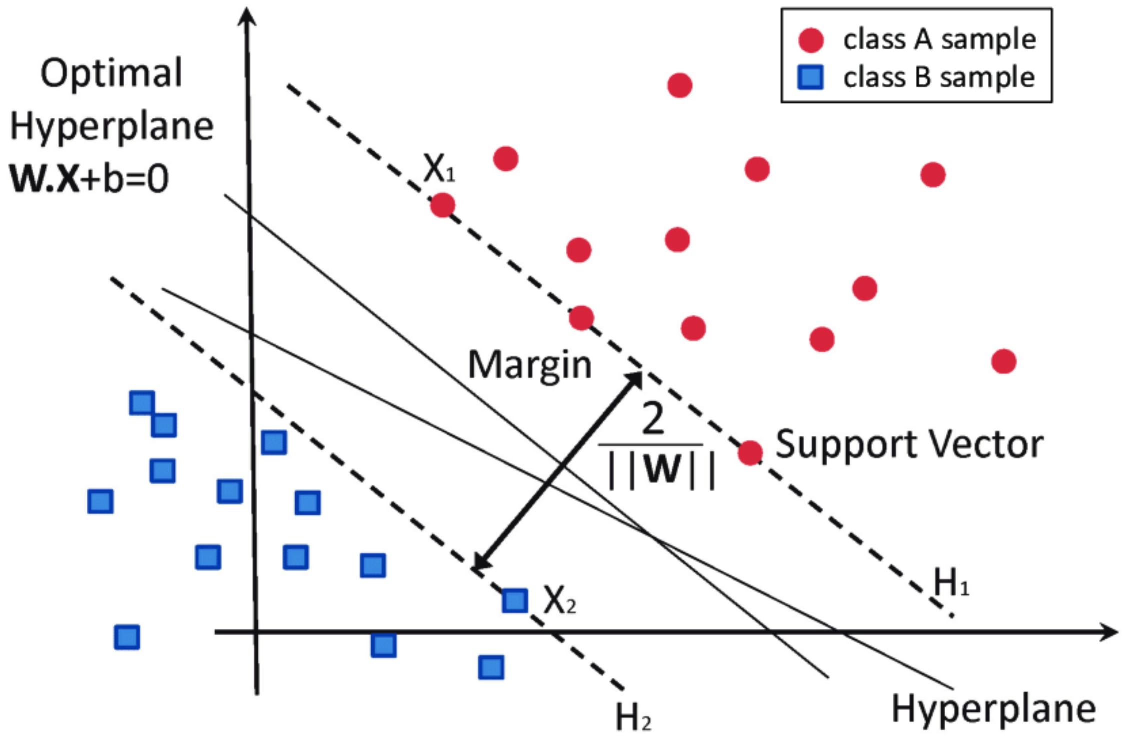

Figure 3: Sample Diagram of Data Points Plotted via SVM algorithm

Diagram by Esperanza García-Gonzalo[...], ResearchGate Publication, June 2016

Intuitively, the further from the hyperplane our data points lie, the more likely they are correctly classified. Support vectors are critical data points nearest to the hyperplane, whose position will alter the dividing hyperplane's position. The margin is the distance between the hyperplane and the nearest data point from either side, so the goal is to output a hyperplane with the greatest possible margin between the hyperplane and any point within the dataset, giving a greater chance of correctly classified new data.

In cases with no obvious hyperplane, kernelling must be done to continuously map data points onto higher dimensions until a distinguishable boundary can be formed to segregate them. (Bambrick, 2016)

Support Vector Machine works really well with clear separation margins, is effective in high dimensional spaces, and in cases where the number of dimensions is greater than the number of samples. The regressor version possesses a computational advantage over regression algorithms by means of a small subset of training points, making it more memory efficient.

On the flip side, it underperforms when the dataset contains overlapping classes, or when the number of features for each data point exceeds the number of training data samples. The final model is difficult to visualise, so making small calibrations is difficult. Visualising SVM's Cost-C and gamma hyperparameters are difficult and hence hard to fine-tune.

Data : Dataset with p^* variables and binary outcome.
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;
Train the SVM model;
$p \leftarrow p^*$;
while $p \geq 2$ **do**
    $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the p variables and observations in **Data**;
    $w_p \leftarrow$ calculate weight vector of the SVM_p $(w_{p1}, \ldots, w_{pp})$;
    $rank.criteria \leftarrow (w_{p1}^2, \ldots, w_{pp}^2)$;
    $min.rank.criteria \leftarrow$ variable with lowest value in $rank.criteria$ vector;
    Remove $min.rank.criteria$ from **Data**;
    $Rank_p \leftarrow min.rank.criteria$;
    p $\leftarrow$ p - 1 ;
end
$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$

Figure 4: Support Vector Machine algorithm Pseudocode

Sanz, Valim, Vegas. "SVM-RFE: Selection and Visualization", Research Gate, Nov 2018

2.3 Random Forest Algorithm (RF)

The Random Forest algorithm is another supervised algorithm consisting of an ensemble of decision trees merged together to obtain a more accurate prediction. A Decision tree is a flowchart like tree structure for decision analysis, each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each terminal leaf node holds a class label. Decision trees implemented in machine learning list all possible alternatives to every question and asks questions in True/False form. (bshyamanth, 2019)

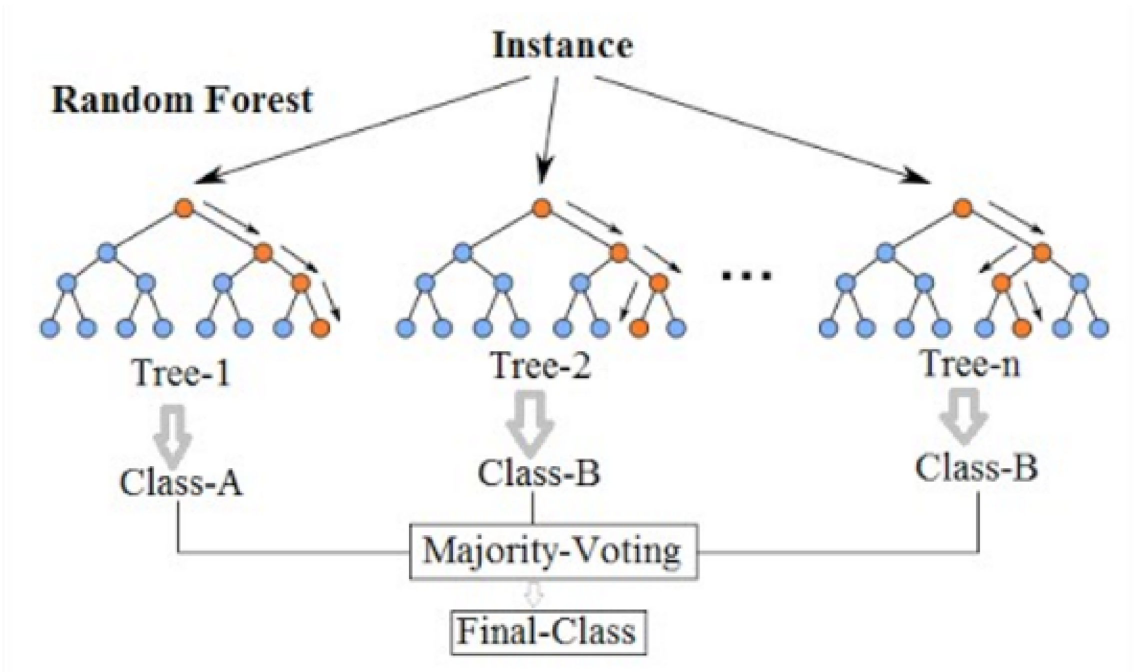

Figure 5: Simplified visual representation of the Random Forest algorithm

Koehrsen, Will. "Random Forest Simple Explanation." Medium, Medium, 18 Aug. 2020

A Random forest combines many decision trees into a single model to increase the information scope. Individually, predictions made by decision trees may not be accurate, but combined together, the variance is filtered out and predictions will be closer to the real thing. Therefore, more trees in the forest will yield higher accuracy. (Donges, 2021)

One its biggest advantage is its versatility, and how easy it is to view the relative importance assigned to each input feature. Like SVM, it can be used for both regression and classification tasks, and the default hyperparameters used often produce good prediction results.

Its primary drawback is that an excessively large number of trees will result in the algorithm model being too slow in creating predictions. They are quick to train, but will be slow in creating predictions. A more accurate model requires more trees, which inevitably results in a slower model. To utilise it in real-world application, the program must be coded to have the right number of trees. Time complexity of the algorithm is important when creating predictions in real time and executing the trades, especially when trading on short time frames.

```
To generate c classifiers:
for i = 1 to c do
   Randomly sample the training data D with replacement to produce D_i
   Create a root node, N_i containing D_i
  Call BuildTree( N_i )
end for

BuildTree(N):
if N contains instances of only one class then
   return
else
   Randomly select x% of the possible splitting features in N
   Select the feature F with the highest information gain to split on
   Create f child nodes of N , N_1 ,..., N_f , where F has f possible values ( F_1 , … , F_f )
   for i = 1 to f do
      Set the contents of N_i to D_i , where D_i is all instances in N that match
      F_i
      Call BuildTree( N_i )
   end for
end if
```

Figure 6: Random Forest algorithm Pseudocode

Sirikulviriya, Sindhu Pinyo. "Integration of Rules from a Random Forest", ACSIT Press, 2011

3 Experiment Methodology

3.1 Experimental Procedure

Python implementations of these two algorithms will be compiled and tested on sufficiently large and diverse datasets of historical stock prices. As previously mentioned, the ADJUSTED CLOSE stock prices will be used as input since they account for stock splits and dividends. Each algorithm will be evaluated based on the feasibility criteria in section 3.3. Detailed analysis of the statistics will provide sufficient information and insight into its feasibility and potential in the financial markets.

Python is the programming language most commonly used by data scientists, machine learning developers, and algorithmic traders today. Its syntax is simplistic, and provides access to a rich set of libraries for all sorts of data science and machine learning applications — `scipy, pandas, sklearn, numpy, keras,and scrapy`. `Scikit-learn` specifically was designed for machine learning and already features various classification, regression, and clustering algorithms, so it will be the library employed in the investigation. It was also designed to work with the Python numerical and scientific libraries `NumPy` and `SciPy`, making Python the most suitable language. (Beklemysheva, 2021)

Programs were compiled in Google Colabotary, an IDE for executing python code through the browser, and well suited for machine learning. Implementation of the Scikit-learn library algorithm functions were taken from published code, but modified to load and pre-process the necessary input, and output the necessary evaluation metrics.

The implementation approach will be to predict the price of a stock 'n' days into the future based on the current adjusted close price. The value of 'n' can be altered as one wishes, but it is set to a default of 30 days. The main program structure was written by (K Dhiraj, 2020) in Medium publications, equipped with the basic functionalities, but are altered to specifically process stock prices and output relevant metrics.

Program Outline:

1. Import necessary packages: pandas, numpy, math, Scikit-learn
2. Load historical stock prices of company in CSV file using pandas data reader
3. Pre-process CSV dataframe into independent (x) and dependent (y) datasets. x dataset being the adjusted close prices for the last 5 years, y dataset holding respective prices 'n' days in the future
4. Split datasets into 80% training data and 20% testing data
5. Train respective algorithms on training data
6. Create predictions from independent (x) testing data
7. Output evaluation metrics

As a control and to ensure a fair experiment, both programs were run on the same computer with the same specifications:

Processor - Intel Core i7-10700K CPU @ 3.80GHz, 8 Cores, 16 Logical Processor(s)

RAM - 32 GB

Operating System - Microsoft Windows 10 Pro

3.2 Datasets

The algorithms must be tested on sufficiently large and varied datasets to produce a quality investigation, and to maintain fairness. The datasets are historical stock prices within the S&P 500, but are obtained from different industries, sectors, and market capitalisations.

The reason for limiting the choice of stocks to those within the S&P 500 is because that stock market index measures the stock performance of 500 large companies listed in the United States. A specific group of stocks needed to be selected for experimentation, and the S&P 500 is a well-known, diversified group with sufficiently large companies. US companies not included in the index are often too risky due to small market capitalisations, low volume, and irrational behaviour from insider trading. Steering clear of precarious stocks will present a clearer picture of the algorithms' performance under standard circumstances.

The stock prices are imported from Yahoo Finance, which features a complete database of historical stock quotes for almost all stocks in the S&P 500 and more. They record the OPEN, HIGH, LOW, CLOSE, ADJUSTED CLOSE, VOLUME in daily, weekly, and monthly timeframes. All quotes are downloadable in comma-separated values (CSV) format, a delimited text file that uses commas as field separators.

The index consists of 11 sectors, which can be further divided into 113 industries. Stocks within the same sector are highly correlated; even more so within specific industries, since they are influenced by similar factors, resulting in them often following the same cyclical trends. Hence, for every industry, one representative stock of the largest market capitalisation and highest trading volume is selected. This way the sectors in which the algorithms perform most optimally can be identified.

Its market capitalisation ranges from $4.2 billion (Mattel, Inc) to $1 trillion (Apple Inc). Larger companies require more trading volume to bring out a price shift as compared to smaller companies (Seth, 2021), so categorising by market capitalisation will determine if company size has a significant impact on the algorithm's performance. Following commonly used categories by Investopedia, all 113 companies can be classified as either Mega-Cap (>$200 billion), Large-Cap ($10 - $200 billion), or Mid-Cap ($2 - $10 billion)

Several tables have been compiled and displayed in the appendix, outlining the datasets compiled from scratch. Quantitative figures were taken from Financial Visualisations. Datasets used will be daily prices from operational market days for the last 5 years starting on April 1st.

3.3 Feasibility Criteria

Determining the algorithms' feasibility will be based on 2 primary criteria - their prediction performance and computational complexity. Prediction performance determines the ability to create profitable trades and is the main determinant of feasibility. The more accurate the predictions the more profitable, and the more consistently the algorithm can create these predictions the more reliable it becomes. The necessary evaluation metrics will be implemented to compare the predictions to the training dataset and provide quantitative data to analyse. Computational complexity is just to do with the resources required to run the algorithms, which will be discussed in a succeeding section.

4 Experimental Results

4.1 Tabular Representation

4.1.1 Overall Performance Metrics

	Support Vector Machine			Random Forest		
	R^2	RMSE	MAE	R^2	RMSE	MAE
Avg	0.81649701	14.5754372	9.15858357	0.768234871	13.1788497	8.85020969
Min	0.39801951	1.15144476	0.85307259	0.242576629	1.41050716	1.0483974
Max	0.98046439	345.617873	189.745969	0.978278769	166.315782	115.716756

4.1.2 Performance Metrics by Market Capitalisation

	Support Vector Machine			Random Forest		
	R^2	RMSE	MAE	R^2	RMSE	MAE
>$200B						
Avg	0.88225719	34.5613232	20.0240981	0.875442291	23.1317825	14.9423483
Min	0.76402355	1.75761121	1.28991308	0.726689698	2.19010008	1.58785367
Max	0.98046439	345.617873	189.745969	0.978278769	158.776077	96.2824388
$10-$200B						
Avg	0.80524302	10.6145254	7.02415266	0.748419866	11.2329558	7.65622816
Min	0.39801951	1.15144476	0.85307259	0.242576629	1.41050716	1.0483974
Max	0.97216737	177.460283	120.871441	0.967526782	166.315782	115.716756
$2-$10B						
Avg	0.75735087	8.76299400	5.57208476	0.704837231	9.68503118	6.77713559
Min	0.44560545	3.03792745	2.11269197	0.369468769	3.50570372	2.55254260
Max	0.92860591	18.3696104	11.0013365	0.907510465	19.5904212	13.4931324

4.1.3 Performance Metrics by Market Sector

	Support Vector Machine			Random Forest		
Market Sector	R^2	RMSE	MAE	R^2	RMSE	MAE
Technology	0.9083905	9.5879830	5.9373416	0.884161	10.295952	6.670239
Utilities	0.8889415	3.2417249	2.3085715	0.829446	3.7033557	2.636495
Industrials	0.8660602	9.2616795	5.9903533	0.822671	10.568090	7.004258
Healthcare	0.8362428	17.246083	11.806360	0.798837	17.710673	12.38906
Cons Cyclical	0.8117181	34.966839	21.274312	0.774679	25.484341	16.85490
Cons Defensive	0.8070184	5.6535652	3.9052380	0.757193	6.2671366	4.466867
Financials	0.7824754	11.393518	7.1324865	0.732030	11.802677	7.861758
Real Estate	0.7774008	6.3188861	4.0399110	0.697622	7.1071367	4.779713
Basic Materials	0.7342145	7.8365365	4.9444413	0.673292	8.3188084	5.645625
Comm Services	0.7252882	29.647747	17.626874	0.665884	22.928595	15.31492
Energy	0.7278824	4.4712189	2.9837944	0.644101	4.9922572	3.559152

*Ranked by order of average R^2 values

4.2 Evaluation Metrics Rationale

Accuracy for regression algorithms must be reported as prediction errors or how close the predictions were to the expected values. The 3 error metrics used were: R^2, RMSE, and MAE because they yield invaluable information about the relationship between the predictions and training dataset. (Brownlee, 2021)

1. Coefficient of Determination (R^2)

   measures how close data points are to the fitted regression line, therefore the R^2 value informs one of the relationship strengths between the model and variable.

2. Root Mean Squared Error (RMSE)

   Is the square root of the average of the squared errors. Due to it being an absolute value and in the same units as the response variable, it makes for a solid measure informing one of how accurately the model predicts the response. Since values are squared before averaged, magnifying large errors, and ignoring very small ones. Thus, it is also Ideal for evaluating feasibility for leveraged trading, where monetary profits/losses are amplified as well.

3. Mean Absolute Error (MAE)

   Is the sum of the absolute value of the differences between all the expected values and predicted values, divided by the total number of predictions. Just like RMSE, it also measures errors in the same units and informs one of the model accuracy, except each individual error values' contribution to the final result is linear. (Eg. Error of 10 contributes twice as much as an error of 5). Hence, it is ideal for evaluating feasibility for non-leveraged stock trading. (Villalobos, 2020)

5 Analysis & Evaluation

5.1 Results Analysis

Overall performance metrics for all 113 stocks show that predictions from algorithms produced predictions with moderately strong positive correlation to the actual prices, yielding average R^2 values of 0.81649 and 0.76823 respectively. Average RMSE and MAE displayed relatively low errors of around 9-14 USD. Those values alone are sufficient to start assembling a system, but the large standard deviation present in all 3 metrics calls for some consideration. R^2 values ranged from as high as 0.98046 / 0.97827, down to 0.39801 / 0.24257. The same is seen for RMSE and MAE values, the worst errors deviating between 114-344 USD.

Both algorithms appeared to produce more strongly correlated predictions for companies with larger market capitalisations. RMSE and MAE followed the same trend, indicating larger errors in larger companies, though it can be attributed to their more expensive stocks. R^2 values in mega-cap had the least standard deviation, ranging from 0.72668 - 0.98046, whilst the 2 smaller categories occasionally produced values below 0.7, the lowest extreme being a weak 0.24257.

From market sector categorisation, the algorithms created the best predictions in Technology, and the worst in Energy. Noticeably large errors were present for Consumer Cyclical, Communication Services, and Healthcare. Though this too may be caused by expensive stock prices.

5.2 Computational Complexity

Measuring algorithm time performance must be with regards to the input size (n), using the Big O Notation. Raw time measurements are misleading, depending heavily on the hardware used. Thus reflects secondary elements, not on the algorithm itself.
Space complexity is the amount of extra memory required to execute said algorithm, also with respect to input. This is important to be taken into consideration since different potential clients interested in running the algorithms are bound to possess different computer specifications. Poor complexities will make them unsuitable for an average investor.

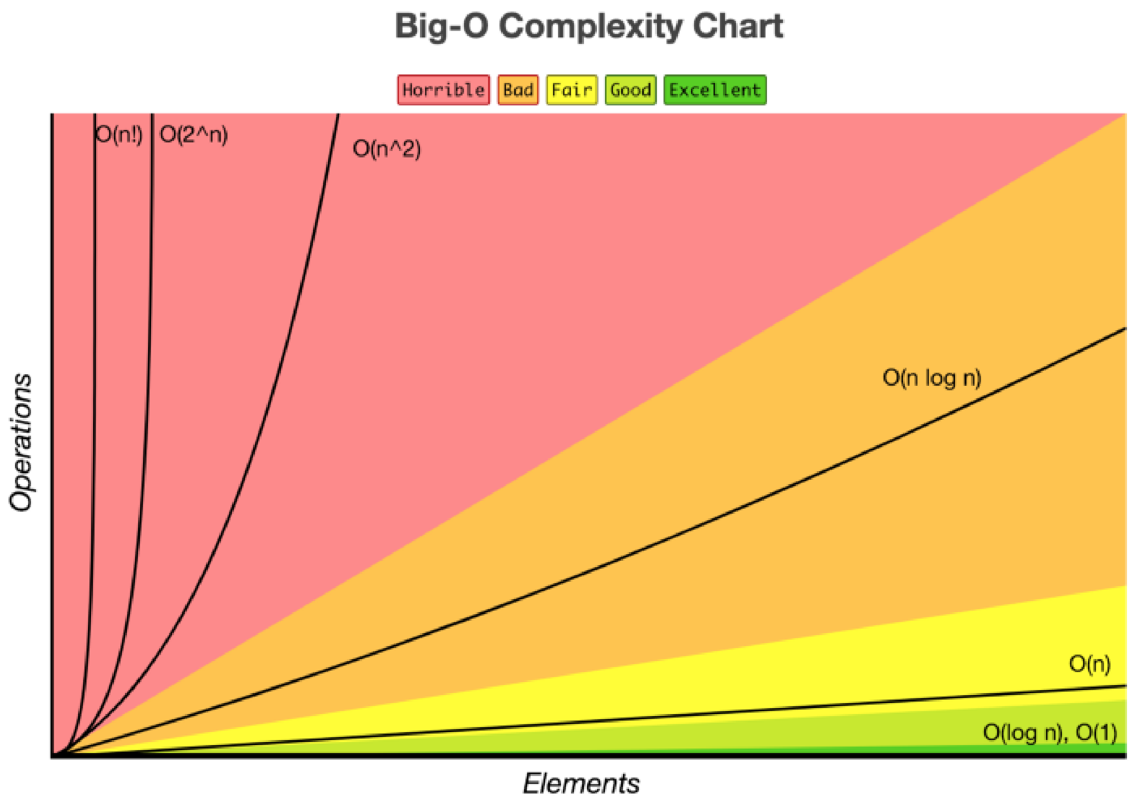

Figure 5: Big O Notation Complexity Chart

Kumar, Paritosh. "Time Complexity of ML Models." Medium, Analytics Vidhya, 31 Mar. 2021,

<u>SVM Complexity</u>

Training Time Complexity = $O(n^2)$

Run-time Complexity = $O(k*d)$ (k = Number of Support Vectors, d = data dimensionality)

Falling under the horrible category for training time, and only fair for run-time, it became clear that SVM should be avoided when input size is exceptionally large.

RF Complexity

Training Time Complexity= O(n*log(n)*d*k) (k=number of Decision Trees)

Run-time Complexity= O(tree depth* k)

Space Complexity= O(tree depth *k)

Training time falls under the bad category, but run-time and space complexity are fair. RF can be executed comparatively faster than other algorithms.

Any complexity drawbacks can be mitigated by utilising advanced hardware. Thus, this poses a problem to its feasibility for an average retail investor as compared to a multinational establishment with plenty of computing power and resources.

5.3 Methodology Limitations & Evaluation

Due to time constraints and limited expertise, the methodology had several limitations to take into consideration. The program implementation predicts the stock price ‘n’ days in the future, 30 days was used but that might be far from the optimal value. The approach implemented was for simplicity’s sake, but different approaches can be used: Including using classifiers, or using more than just adjusted close prices as input parameters. 5 years worth of daily adjusted close prices was used. Again, there is an optimal size of training data to be used to train a model which would yield the best results, one not too large or too small to prevent overfitting/underfitting. These optimal values and approach feasibility can only be determined from further experimentation.

Regarding the dataset used, only selected USA stocks in the S&P 500 were utilised for this investigation. Further testing should be done on different timeframes (weekly and monthly), more than 1 representative company for each industry, and stocks from different countries/indices.

5.4 Final Conclusion

This research paper aimed to assess the feasibility of using machine learning algorithms in stock price prediction. After rigorous experimentation and analysis, this paper concludes that using the Support Vector Machine and Random Forest algorithms alone are NOT FEASIBLE for consistently creating accurate predictions used to reliably execute profitable trades on companies with market capitalisations upwards of 7 Billion USD. (Mitchell, 2020) As of now, these algorithms alone; with their high level of predictive standard deviation, large margin errors, (especially in smaller companies which are more preferably traded due to higher volatility) and mediocre computational complexities pose the risks of executing loss-making trades at the cost of time and computational resources, and are thus, nowhere near feasible for everyday retail traders, let alone multinational establishments that utilise leveraged trading. The potential for outperforming the S&P 500 yearly return of 10% is unlikely with R^2 values below 0.5 is highly unlikely. (Kierski, 2017)

More experimentation and research could possibly be carried out to integrate other types of algorithms such as Natural language processing or traditional technical indicators, in order to construct a sophisticated enough trading system reliable and accurate enough to consistently generate profits; but no matter how sophisticated the compiled program may be or how advanced any machine learning algorithm becomes, ultimately fluctuations in the stock market are driven by the emotional sentiment of traders. No technology in the world could possibly be able to perfectly predict the emotional sentiments of millions of traders trading every millisecond. Though there is a statistical certainty that some model existing out there is capable of predicting prices well enough, at the end of the day the stock market is a zero-sum game, someone's gain is another person's loss. Anyone with a decent enough model would stand to gain by not sharing it with anybody else. Just like traditional stock investing strategies, their performance decreases as more people employ it. Therefore great models tend to not be shared, and have an almost negligible chance of breaking the market.

6 Works Cited

[1] "AI." Brown Armadillo, 2012 www.brownarmadillo.com/ai/.

[2] Brownlee, Jason. "Difference Between Classification and Regression in Machine Learning." Machine Learning Mastery, 21 May 2019, machinelearningmastery.com/classification-versus-regression-in-machine-learning/.

[3] Brownlee, Jason. "Regression Metrics for Machine Learning." Machine Learning Mastery, 15 Feb. 2021, machinelearningmastery.com/regression-metrics-for-machine-learning/.

[4] Burns, Ed. "What Is Machine Learning (ML)?" SearchEnterpriseAI, TechTarget, 24 June 2020, searchenterpriseai.techtarget.com/definition/machine-learning-ML.

[5] Bushkovskyi, Oleksandr. Machine Learning Algorithms: 4 Types You Should Know, The App Solutions, 7 Dec. 2018,theappsolutions.com/blog/development/machine-Learning-algorithm-types/.

[6] Butcher, Sarah. "JPMorgan's Massive Guide to Machine Learning Jobs in Finance." EFinancialCareers, 5 May 2018,www.efinancialcareers.co.uk/news/2017/12/machine-learning-and-big-data-j-p-morgan?_ga=2.171766117.413759266.1612972670-1180954039.1612972670.

[7] bshyamanth, 17 Apr. 2019"Decision Tree." GeeksforGeeks, www.geeksforgeeks.org/decision-tree/.

[8] Donges, Niklas. July 22, 2021. "A Complete Guide to the Random Forest Algorithm." Built In, builtin.com/data-science/random-forest-algorithm.

[9] Info@lawtomated.com. "Supervised Learning vs Unsupervised Learning. Which Is Better?" Lawtomated, 2 Feb. 2020, lawtomated.com/supervised-vs-unsupervised-learning-which-is-better/.

[10] K, Dhiraj. "Top 4 Advantages and Disadvantages of Support Vector Machine or SVM." Medium, Medium, 26 Dec. 2020,dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107.

[11] Koehrsen, Will. "Random Forest Simple Explanation." Medium, Medium, 18 Aug. 2020, williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d.

[12] Kumar, Paritosh. “Time Complexity of ML Models.” Medium, Analytics Vidhya, 31 Mar. 2021, medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770.

[13] Lv, Dongdong, et al. “An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy.” Mathematical Problems in Engineering, Hindawi, 14 Apr. 2019, www.hindawi.com/journals/mpe/2019/7816154/#introduction.

[14] “Machine Learning for Trading - Topic Overview.” Sigmoidal, 3 Nov. 2017, sigmoidal.io/machine-learning-for-trading/#:~:text=Machine%20Learning%20offers%20the%20number,can%20monitor%20and%20respond%20to.

[15] “Machine Learning in Finance - Overview, Applications.” Corporate Finance Institute, 28 July 2020, corporatefinanceinstitute.com/resources/knowledge/other/machine-learning-in-finance/#:~:text=In%20finance%2C%20machine%20learning%20algorithms,outcomes%20without%20being%20explicitly%20programmed.

[16] Medium, N.D heartbeat.fritz.ai/random-forest-regression-in-python-using-scikit-learn-9e9b147e2153.

[17] Medium, heartbeat.fritz.ai/support-vector-regression-in-python-using-scikit-learn-89cc18e933b7.

[18] Mitchell, Cory. “Make Money Day Trading Despite the Challenges.” The Balance, 12 Feb. 2020, www.thebalance.com/why-it-is-so-hard-to-make-consistent-money-day-trading-1031238

[19] “The Profitability of Day Traders.” Taylor & Francis, www.tandfonline.com/doi/abs/10.2469/faj.v59.n6.2578.

[20] Seth, Shobhit. “Basics of Algorithmic Trading: Concepts and Examples.” Investopedia, Investopedia, 28 Aug. 2020, www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp.

[21] Seth, Shobhit. “Market Capitalization Defined.” Investopedia, Investopedia, 29 Jan. 2021, www.investopedia.com/investing/market-capitalization-defined/.

[22] “Support Vector Machines: A Simple Explanation.” KDnuggets, www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html#:~:text=A%20Support%20Vector%20Machine%20(SVM,both%20classification%20and%20regression%20purposes.&text=SVMs%20are%20based%20on%20the,shown%20in%20the%20image%20below.

[23] Venables, Michelle. “Quantitative, Qualitative or Maybe Both?” Medium, Towards Data Science, 24 Jan. 2020, towardsdatascience.com/quantitative-qualitative-or-maybe-both -542553d09918.

[24] Villalobos, Juan Orozco. “Mean Absolute Error vs Root-Mean Square Error.” BrainsToBytes, BrainsToBytes, 29 Jan. 2020 ,www.brainstobytes.com/mean-absolute-error-vs-root-mean-square-error/.

[25] “Why Use Python for AI and Machine Learning?” SteeKiwi, 12 Jan. 2021, steelkiwi.com/blog/python-for-ai-and-machine-learning/#:~:text=Benefits%20that %20make%20Python%20the,overall%20popularity%20of%20the%20language.

7 Appendix

7.1 Program Code

```
[ ] # Package imports
    import datetime
    import math
    from math import sqrt
    import numpy as np
    import pandas as pd
    import pandas_datareader as web
    from sklearn.metrics import r2_score, mean_squared_error
    from sklearn.metrics import mean_absolute_error as mae
    from sklearn.svm import SVR
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.model_selection import train_test_split
```

```
[ ] # Loading CSV file
    data = web.DataReader('AAPL', data_source='yahoo', start='2016-04-01', end='2021-04-01')

    # Check data has loaded properly
    print(data.head())
    print(data.shape)
```

```
# Formatting and Manipulating Loaded CSV Data

# Create new dataframe with only Adjusted Close Column
data = data[['Adj Close']]
print(data.head())
print()
# A variable for predicting 'n' days out into the future
forecast_out = 30
# Create another column shifted 'n' units up,
# for storing predicted price values 'n' days into the future.
data['Prediction'] = data[['Adj Close']].shift(-forecast_out)
# Print the new data set
print(data.tail())
print()

# Create the independent data set (x)
# Convert the dataframe to a numpy array
x = np.array(data.drop(['Prediction'],1))
#Remove the last '30' rows
x = x[:-forecast_out]
print('Independent Dataset (x): ')
print(x)
print()

# Create the dependent data set (y)
# Convert the dataframe to a numpy array
y = np.array(data['Prediction'])
# Get all of the y values except the last '30' rows
y = y[:-forecast_out]
print('Dependent Dataset (y): ')
print(y)

# Splitting into 80% training and 20% testing data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

```
# Create and Train the Support Vector Machine Regressor
# Used radial basic function (default kernal model for SVR)
svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)
svr_rbf.fit(x_train, y_train)

# Make SVM Prediction
SVM_y_pred = svr_rbf.predict(x_test)
print('Support Vector Machine Predicition: ')
print(SVM_y_pred)

# Create and Train the Random Forest Regressor
rand_for = RandomForestRegressor(random_state=42, n_estimators=300)
rand_for.fit(x_train, y_train)

# Make RF Prediction
RF_y_pred = rand_for.predict(x_test)
print('Random Forest Prediction: ')
print(RF_y_pred)
```

```
[ ]  # Support Vector Machine Evaluation
     print ('Support Vector Machine Metrics: ')

     # SVM confidence returns the coefficient of determination R^2 of the prediction.
     # R^2 is a statistical measure that represents the goodness of fit of a regression model.
     # The best possible score is 1.0
     svm_r2 = r2_score(y_test, SVM_y_pred)
     print("Coefficient of Determination/Confidence (R² Value): ", svm_r2)

     # RMSE is how close a fitted line is to data points.
     # The lower to better
     mse = mean_squared_error(y_test, SVM_y_pred)
     rmse = np.sqrt(mse)
     print("Root Mean Squared Error: ", rmse)

     # Mean Absolute Error shows the average difference between the actual data value
     # and the value predicted by the model
     print("Mean Absolute Error: ", mae(y_test, SVM_y_pred))

     print('')

     # Random Forest Evaluation
     print('Random Forest Metrics: ')

     # SVM confidence returns the coefficient of determination R^2 of the prediction.
     # R^2 is a statistical measure that represents the goodness of fit of a regression model.
     # The best possible score is 1.0
     rf_r2 = r2_score(y_test, RF_y_pred)
     print("Coefficient of Determination Confidence (R² Value): ", rf_r2)

     # RMSE is how close a fitted line is to data points.
     # The lower to better
     mse = mean_squared_error(y_test, RF_y_pred)
     rmse = np.sqrt(mse)
     print("Root Mean Squared Error: ", rmse)

     # Mean Absolute Error shows the average difference between the actual data value
     # and the value predicted by the model
     print("Mean Absolute Error: ", mae(y_test, RF_y_pred))
```

7.2 Dataset Tables

TABLES 1-11

Categorisation by Sector & Industry

TABLE 1		
TECHNOLOGY (27.48%)		
Industry	**Representative Company**	**Ticker**
Communication Equipment	Cisco Systems	CSCO
Computer Hardware	HP	HPQ
Consumer Electronics	Apple	AAPL
Electronic Components	TE Connectivity	TEL
Information Technology Services	Accenture plc	ACN
Scientific & Technical Instruments	Keysight Technologies	KEYS
Semiconductor Equipment & Materials	Applied Materials	AMAT
Semiconductors	NVIDIA	NVDA
Application Software	salesforce.com	CRM
Infrastructure Software	Microsoft	MSFT
Solar	Enphase Energy	ENPH

TABLE 2		
COMMUNICATION SERVICES (10.90%)		
Industry	**Representative Company**	**Ticker**
Advertising Agencies	Omnicom Group	OMC
Broadcasting	Fox Corporation	FOXA
Electronic Gaming & Multimedia	Activision Blizzard	ATVI
Entertainment	Walt Disney Company	DIS
Internet Content & Information	Alphabet	GOOG
Telecom Services	Verizon Communications	VZ

TABLE 3		
HEALTHCARE (14.58%)		
Industry	**Representative Company**	**Ticker**
Biotechnology	Regeneron Pharmaceuticals	REGN
Diagnostics & Research	Thermo Fisher Scientific	TMO
General Drug Manufacturers	Johnson & Johnson	JNJ
Specialty & Generic Drug Manufacturers	Zoetis	ZTS
Healthcare Plans	UnitedHealth Group	UNH
Health Information Services	Cerner	CERN
Medical Care Facilities	HCA Healthcare	HCA
Medical Devices	Abbott Laboratories	ABT
Medical Distribution	Mckesson Corporation	MCK
Medical Instruments & Supplies	Intuitive Surgical	ISRG
Pharmaceutical Retailers	Walgreens Boots Alliance	WBA

TABLE 4		
UTILITIES (3.13%)		
Industry	**Representative Company**	**Ticker**
Diversified Utilities	Dominion Energy	D
Independent Power Producers	NRG Energy	NRG
Regulated Electric	NextEra	NEE
Regulated Gas	Centerpoint Energy	CNP
Regulated Water	American Water Works	AWK

TABLE 5		
FINANCIALS (9.89%)		
Industry	**Representative Company**	**Ticker**
Asset Management	BlackRock	BLK
Diversified Banks	Bank of America	BAC
Regional Banks	Truist Financial Corp	TFC
Capital Markets	Morgan Stanley	MS
Credit Services	Visa	V
Financial Data & Stock Exchanges	S&P Global	SPGI
Insurance Brokers	Marsh & McLennan	MMC
Diversified Insurance	Berkshire Hathaway	BRK-B
Life Insurance	MetLife	MET
Property & Casualty Insurance	The Progressive Corp	PGR
Reinsurance Insurance	Everest Re Group	RE
Specialty Insurance	Assurant	AIZ

TABLE 6		
CONSUMER CYCLICAL/DISCRETIONARY (11.18%)		
Industry	**Representative Company**	**Ticker**
Apparel Manufacturing	V.F. Corporation	VFC
Apparel Retail	The TJX Companies	TJX
Auto Manufacturers	General Motors	GM
Auto Parts	Aptiv PLC	APTV
Auto & Truck Dealerships	CarMax	KMX
Footwear & Accessories	NIKE	NKE
Furnishings, Fixtures, & Appliances	Mohawk Industries	MHK
Home Improvement Retail	The Home Depot	HD
Internet Retail	Amazon	AMZN
Leisure	Pool Corporation	POOL
Lodging	Marriott International	MAR
Luxury Goods	Tapestry	TPR
Packaging & Containers	Ball Corporation	BLL
Personal Services	Rollins	ROL
Residential Construction	D.R Horton	DHI
Resorts & Casinos	Las Vegas Sands Corp	LVS
Restaurants	McDonald's Corporation	MCD
Specialty Retail	Best Buy Co	BBY
Travel Services	Booking Holdings	BKNG

TABLE 7		
INDUSTRIALS (7.90%)		
Industry	**Representative Company**	**Ticker**
Aerospace & Defense	Boeing	BA
Airlines	Delta Air Lines	DAL
Building Products & Equipment	Carrier Global	CARR
Business Equipment & Supplies	Avery Dennison	AVY
Consulting Services	IHS Markit	INFO
Engineering & Construction	Johnson Controls	JCI
Farm & Heavy Construction Machinery	Caterpillar	CAT
Industrial Distribution	Fastenal Company	FAST
Integrated Freight & Logistics	United Parcel Service	UPS
Railroads	Union Pacific	UNP
Rental & Leasing Services	United Rentals	URI
Security & Protection Services	Allegion	ALLE
Specialty Business Services	Global Payments	GPN
Specialty Industrial Machinery	General Electric	GE
Staffing & Employment Services	Automatic Data Processing	ADP
Tools & Accessories	Stanley Black & Decker	SWK
Trucking	Old Dominion Freight Line	ODFL
Waste Management	Waste Management Inc	WM

TABLE 8		
CONSUMER DEFENSIVE/STAPLES (7.05%)		
Industry	**Representative Company**	**Ticker**
Beverage Brewers	Molson Coors	TAP
Non-Alcoholic Beverages	Coca-Cola	KO
Wineries & Distilleries	Constellation Brands	STZ
Confectioners	Mondelez International	MDLZ
Discount Stores	Costco Wholesale	COST
Farm Products	Tyson Foods	TSN
Food Distribution	Sysco	SYY
Grocery Stores	The Kroger	KR
Household & Personal Products	Procter & Gamble	PG
Packaged Foods	Kraft Heinz	KHC
Tobacco	Altria Group	MO

TABLE 9		
REAL ESTATE (2.80%)		
Industry	**Representative Company**	**Ticker**
Real Estate Services	CBRE Group	CBRE
REIT Healthcare Facilities	Welltower	WELL
REIT Hotel & Motel	Host Hotels & Resorts	HST
REIT Industrial	Prologis	PLD
REIT Office	Digital Realty Trust	DLR
REIT Residential	Equity Residential	EQR
REIT Retail	Simon Property Group	SPG
REIT Specialty	Crown Castle International	CCI

TABLE 10		
BASIC MATERIALS (2.56%)		
Industry	**Representative Company**	**Ticker**
Agricultural Inputs	The Mosaic	MOS
Building Materials	Vulcan Materials	VMC
Chemicals	Air Products & Chemicals	APD
Copper	Freeport-McMoRan	FCX
Gold	Newmont	NEM
Specialty Chemicals	Linde	LIN
Steel	Nucor Corporation	NUE

TABLE 11		
ENERGY (2.53%)		
Industry	**Representative Company**	**Ticker**
Oil & Gas E&P	ConocoPhillips	COP
Oil & Gas Equipment & Services	Baker Hughes	BKR
Oil & Gas Integrated	Exxon Mobil	XOM
Oil & Gas Midstream	Kinder Morgan	KMI
Oil & Gas Refining & Marketing	Marathon Petroleum	MPC

Tables 12

Categorisation by Market Capitalisation

*All quantitative figures are accurate as of 17-02-2021

Company	Ticker	Market Cap (B)	Category
Apple	AAPL	2268.58	Mega-Cap
Microsoft	MSFT	1844.00	
Amazon	AMZN	1642.69	
Alphabet	GOOG	1444.40	
Berkshire Hathaway	BRK-B	589.18	
Visa	V	450.24	
Johnson & Johnson	JNJ	437.16	
NVIDIA	NVDA	377.61	
Walt Disney	DIS	346.36	
Unitedhealth Group	UNH	316.02	
Procter & Gamble	PG	316.01	
Home Depot	HD	298.28	
Bank of America	BAC	284.44	
Salesforce.com	CRM	228.97	
Verizon Communications	VZ	225.73	
NIKE	NKE	225.64	
Abbott Laboratories	ABT	224.44	
Exxon Mobil	XOM	217.08	
Coca-Cola	KO	214.37	
Cisco Systems	CSCO	197.73	Large-Cap
Thermo Fisher Scientific	TMO	192.82	
Accenture PLC	ACN	170.94	

NextEra Energy	NEE	160.63	Large-Cap
McDonald's	MCD	160.50	
Costco Wholesale	COST	156.56	
Union Pacific	UNP	139.35	
United Parcel Service	UPS	138.94	
Morgan Stanley	MS	135.95	
Linde PLC	LIN	131.13	
Boeing	BA	126.42	
BlackRock	BLK	111.16	
Caterpillar	CAT	110.15	
Applied Materials	AMAT	104.82	
General Electric	GE	102.44	
Intuitive Surgical	ISRG	92.19	
Booking Holdings	BKNG	89.81	
The TJX Companies	TJX	82.26	
S&P Global	SPGI	81.42	
Altria Group	MO	80.51	
Activision Blizzard	ATVI	79.14	
Prologis	PLD	78.85	
Mondelez International	MDLZ	78.18	
Zoetis	ZTS	77.97	
General Motors	GM	76.89	
Truist Financial	TFC	73.31	
Automatic Data Processing	ADP	71.01	
Crown Castle	CCI	68.81	
Global Payments	GPN	59.61	
HCA Healthcare	HCA	59.57	

Dominion Energy	D	58.99	Large-Cap
Marsh & McLennan Companies	MMC	58.07	
Air Products & Chemicals	APD	57.06	
Regeneron Pharmaceuticals	REGN	52.02	
ConocoPhillips	COP	51.38	
Metlife	MET	50.08	
The Progressive Corporation	PGR	49.91	
Waste Management	WM	47.57	
Freeport McMoRan	FCX	47.43	
Newmont	NEM	45.99	
Constellation Brands	STZ	44.03	
Kraft Heinz	KHC	43.42	
Las Vegas Sands	LVS	43.09	
TE Connectivity	TEL	42.71	
Walgreens Boots Alliance	WBA	42.24	
Marriott International	MAR	41.37	
Aptiv PLC	APTV	41.25	
Digital Realty Trust	DLR	39.75	
IHS Marikt	INFO	39.10	
Sysco	SYY	38.98	
Johnson Controls International	JCI	37.68	
Simon Property Group	SPG	36.09	
HP	HPQ	34.89	
Kinder Morgan	KMI	33.70	
Marathon Petroleum	MPC	33.02	
Carrier Global	CARR	32.45	
V.F. Corporation	VFC	32.07	

Best Buy Co	BBY	31.03	Large-Cap
American Water Works	AWK	29.86	
D.R. Horton	DHI	29.16	
Ball Corporation	BLL	29.12	
McKesson	MCK	28.70	
Delta Air Lines	DAL	28.10	
Welltower	WELL	27.59	
Keysight Technologies	KEYS	27.54	
Stanley Black & Decker	SWK	27.37	
Fastenal	FAST	26.60	
The Kroger	KR	25.21	
Equity Residential	EQR	25.11	
Old Dominion Freight Line	ODFL	24.18	
Tyson Foods	TSN	23.88	
Baker Hughes	BKR	23.87	
Enphase Energy	ENPH	23.57	
CBRE Group	CBRE	23.34	
Cerner	CERN	22.59	
Vulcan Materials	VMC	20.94	
United Rentals	URI	20.33	
CarMax	KMX	20.16	
Rollins	ROL	18.14	
Fox Corporation	FOXA	18.09	
Nucor Corporation	NUE	16.85	
Avery Dennison	AVY	14.62	
Omnicom Group	OMC	14.12	
Pool Corporation	POOL	13.07	

Centerpoint Energy	CNP	11.66	Large-Cap
Mohawk Industries	MHK	11.49	
Tapestry	TPR	11.08	
The Mosaic	MOS	11.06	
Host Hotels & Resorts	HST	10.79	
Allegion	ALLE	10.18	
NRG Energy	NRG	9.88	Mid-Cap
Molson Coors Beverages	TAP	9.79	
Everest Re Group	RE	9.67	
Assurant	AIZ	7.54	