**Data 670 Data Analytics**

***Brian Richardson***

**Professor Saleeb Hany**

**Assignment 6**

**08/5/2024**

**Executive Summary**

One area of applications of machine learning is stock market analysis. Stock predictions using machine learning algorithms can help increase profits and reduce losses. The traditional approach to investing involves a lot of time spent doing research and calculations for each investment decision. Using a machine learning algorithm to do some of this research and calculation can significantly reduce the amount of time spent making an investment decision.

An example of a general investing rule for predicting stock prices is that a low P/E increases the chance of the price rising over a long period of time. The lower the P/E the more the increase. The theory is that this is the result of investors favoring low P/E ratios and driving up the stock price over a long period of time by purchasing more of the stock.

Machine learning can verify if the data supports this rule, other rules, or discover a new rule that can be used for price prediction. Then machine learning can be used to predict stock prices using market data. By building good machine learning models, stock data can be input, and a price prediction calculated with a margin of error.

Being able to predict stock prices using machine learning will reduce the amount of time spent investing. It would also allow investors to make better investing decisions. This would mean that there would be a reduction in expenses, time, and losses. Hopefully, this would also mean that there would be an increase in profit.

Table of Contents

[Project Scope (Updated/revised from Assignment 1) 3](#_Toc30340425)

[Problem Description 3](#_Toc30340426)

[Business Understanding 3](#_Toc30340427)

[Organization 4](#_Toc30340428)

[Stakeholders 4](#_Toc30340429)

[Define Business Area 5](#_Toc30340430)

[Business Objectives 5](#_Toc30340431)

[Business Success Criteria 5](#_Toc30340432)

[Background 6](#_Toc30340433)

[Research 6](#_Toc30340434)

[Gaps in this Problem Resolution 6](#_Toc30340435)

[Proposed Project 7](#_Toc30340436)

[Key Performance Indicators 7](#_Toc30340437)

[Project Insights of your Data Analysis 8](#_Toc30340438)

[Project Milestones 8](#_Toc30340439)

[Completion History 8](#_Toc30340440)

[Lessons Learned 9](#_Toc30340441)

[Data Set Description 11](#_Toc30340442)

[High-Level Data Diagram 11](#_Toc30340443)

[Data Definition/Data Profile 11](#_Toc30340444)

[Data Preparation/Cleansing/Transformation 13](#_Toc30340445)

[Data Preparation 1](#_Toc30340446)3

[Data Cleansing 1](#_Toc30340447)4

[Data Transformation 13](#_Toc30340448)

[Data Analysis 1](#_Toc30340449)5

[Data Visualization 1](#_Toc30340450)7

[Data Visualization 1 1](#_Toc30340451)7

[Data Visualization 2 1](#_Toc30340452)8

[Data Visualization 3](#_Toc30340453) 20

[Proposed Visualizations 21](#_Toc30340454)

[Predictive Models](#_Toc30340455) 22

[Predictive Model 1](#_Toc30340456) 22

[Predictive Model 2](#_Toc30340457) 23

[Predictive Model 3](#_Toc30340458) 25

[Predictive Model Review](#_Toc30340459) 28

[Final Results](#_Toc30340460) 29

[Analysis Justification](#_Toc30340461) 29

[Findings](#_Toc30340462) 30

[Review of Success](#_Toc30340463) 31

[Recommendations for Future Analysis](#_Toc30340464) 32

[References](#_Toc30340465) 33

# Project Scope (Updated/revised from Assignment 1)

# Problem Description

There are many different strategies for successful investing. One thing that was common with many of these strategies is that they are almost always not supported by data. For example, one general rule for predicting stock prices is that a low P/E increases the chance of the price rising over a long period of time. The lower the P/E the more the increase. The theory is that this is the result of investors favoring low P/E ratios and driving up the stock price over a long period of time by purchasing more of the stock.

The traditional investment process is about 80%-90% a waste of time. Investors spend 80%-90% of their time looking at stocks that aren’t worth an investment. With modern technology these tasks can be done using machine learning. Using the combination of machine learning algorithms and past stock data a machine learning model can be created to predict future stock prices. Using machine learning we can ascertain if the data supports the rule mentioned above, other common rules, or discover a new rule that can be used for price prediction. Then we can use machine learning to predict stock prices using market data. By building good machine learning models, stock data can be input, and a price prediction calculated with a margin of error.

This project would benefit any investors who spend a lot of time deciding on which stock to buy. Many investing strategies involve using key attributes like P/E ratio. This project should reveal which key attributes are the best for predicting stock prices and help investors who are uncertain about which strategy is best. This would also automate the more tedious parts of the investing process by delegating some tasks to machine learning algorithms. This would not only save time, but hopefully result in more profit.

**The data analytics problem that I am analyzing is using machine learning, with market data to predict stock prices.**

# Business Understanding

There are about 1 million core investment professionals and several million investors in the United States alone. These include asset owners, asset managers (independent wealth management firms), and intermediaries (providers of investment services). Addition stake holders include government regulation agencies such as U.S. Securities and Exchange Commission, and stock exchanges such as the New York Stock Exchange. Since capital is vital to the construction of organizations, the investment industry influences all other industries to varying degrees.

There are several general objectives in the investment industry. They are in no particular order; maximize profit, reduce risk, facilitate growth, capital appreciation, increase income, increase tax efficiency, liquidity, and speculation. To achieve success as an investor, organizations must meet the mentioned business objectives. There is another success criterion that isn’t mentioned but is a part of every business objective. That is to improve efficiency by reducing the time it takes to achieve those business objectives.

Since most organizations are directly or indirectly involved with the investment industry the objectives previously mentioned affect almost every organization. For example, if an investment firm is inefficient, takes serious risk, and doesn’t maximize profit, other organizations might find themselves in a situation where they lack the capital to achieve their goals. Capital is one of the most important requirements for growth (Ross, 2024). Therefore a investor or investment firm that is able to invest efficiently, take less risk, and maximize profit will have more capital to facilitate growth in the economy.

### Organization

This problem affects individual investors, wealth management firms, and investment services providers. There are many individual investors who just can’t invest the time required to make good investment decisions. As a result, they will often use an investment service provider. An investment service provider is an organization that connects wealth management firms with individual investors. Sometimes the wealth management firm and the investment service provider are one organization, but that is not always the case.

Wealth or asset management firms include mutual funds, exchange traded funds, capital group, specialized funds, unit investment trusts, hedge funds, management companies, bonds, stocks, and private equity firms. In general, they own the assets, and offer them to investment service providers who then sell them to individual investors. Some wealth management firms focus on one area, while others have a wide variety of investment products. This means that this problem only affects those firms or individuals that have market funds, or stock funds.

When it comes to stocks there are two ways to invest. The first is to purchase the stock from the company via a broker, and second is to purchase a pooled investment fund such as a mutual fund. Many wealth management firms, and investment service providers offer both as one of their many options. Therefore, this problem will affect most investment organizations to some degree.

### Stakeholders

The three stakeholders are individual investors, investment service providers, and wealth management firms. Individual investors are investors who aren’t necessarily part of any investment organization. They are the customers of the other two stakeholders. Although they usually invest in smaller amounts, they are numerous in number. The individual investor usually cannot devote the proper amount of time to make good investment decisions and usually pays an investment service provider for that service. Therefore, the individual investor could benefit from a machine learning system that could do more time-consuming investment planning.

Investment service providers are organizations that sell investment products to individual investors. Many investment service providers act as the middleman, and they rarely own any of the assets themselves. They often sell advice as well as investment products. Investment service providers could benefit from a machine learning system that could assist them with recommending the best investments.

Wealth management firms usually own the assets and allow others to purchase those assets in the form of different investment products. Some wealth management firms are also investment service providers. While others just provide their investments for other investment service providers to sell. Only those that sell stocks, or stock derived funds would benefit from a machine learning system that could optimize the purchasing of stocks.

# Define Business Area

Individual companies sell a share of their company in the form of a stock. Stocks are often organized by markets. This stock is bought by individuals and other companies. These companies can combine these stocks into investment funds, which they can then sell to individuals. These investments funds are sometimes organized by markets. The business area stretches from the stock supplier to the individual investor and includes every organization that buys the stock regardless of its purpose.

### Business Objectives

The first objective is to maximize profit. The goal of every investment is to produce wealth for the investor. When it comes to stocks, that means buying the stock at a low price and selling it at a higher price. Ideally the stock should be sold at a much higher price. Being able to predict the price is vital to knowing when to sell the stock to maximize profit. Therefore, predicting the price is vital to maximizing profit.

The second objective is to reduce risk. The risk of loss is always present when investing in stock. Loss can occur when a stock is sold for a lower price than when it was purchased, or the company goes bankrupt. In both cases, the stock price would decline for some time. Although it is impossible to predict bankruptcy, a machine learning algorithm that can predict if a stock price will drop can help reduce the risk of loss. Therefore, predicting the price is vital to reducing risk.

The third objective is to reduce the time it takes to analyze stock data. The process of attempting to predict stock prices involves analyzing a lot of data. Humans aren’t very good at analyzing data and are generally much slower compared to computers. Training a computer to do the more difficult data analysis tasks would reduce the time it takes to analyze stock data. Using machine learning a computer can be trained to recognize patterns in stock data and make recommendations based on that data. It would be doing the exact same thing that expert investors do, but much quicker.

### Business Success Criteria

The first success factor is good data. This means having an adequate number of records. Each record in the data represents a date, the days are recorded in order, and there is a change over time. That means that too many missing values could result in different time frames which could result in data that is too noisy to be useful.

The second success factor is using good indicators. There are many indicators that are used by experts to try and predict stock prices. It is very unlikely that all of them are equal. Using inferior indicators will create poor results.

The third success factor is using the correct machine learning model. For this kind of analysis, the following models are recommended; linear regression, random forest, Naïve Bayesian classifier, support vector machines, K-nearest neighbor, ARIMA, recurrent neural networks, long short-term memory, and graph neural networks (Stefano, 2023). These models are very different from each other and will most likely have different levels of success. Choosing the best models is necessary for success.

# Background

There are many projects on Kaggle that have done stock price prediction. All the ones I have looked at predicted the stock price using attributes such as past prices, averages, and volume. I was unable to find any projects that used other indicators such as P/E ratios to predict stock prices. Since experts rarely try to predict stock prices using only the price and volume as indicators this is a bit of an oversight. These projects on Kaggle use a variety of machine learning algorithms with the most common being ARIMA, linear regression, and long short-term memory.

### Research

Using machine learning to predict stock prices is not new. Many algorithms have been used including linear regression, random forest, Naïve Bayesian classifier, support vector machines, K-nearest neighbor, ARIMA, recurrent neural networks, long short-term memory (LSTM), and graph neural networks (Stefano, 2023). The ARIMA model shows some promise in predicting stock prices (Chauhan, 2021). Additionally, the long short-term memory (LSTM) has also been used successfully to predict stock prices (Saya, 2023). Lastly the linear regression and K-Nearest Neighbors model have also been shown to be successful (Paramartha, 2022).

### Gaps in this Problem Resolution

Many of the projects that attempt to predict stock market prices have used historical prices to predict future prices. Some projects have included Natural Language Processing to combine stock prices with sentiment analysis from financial news. Although both are interesting methods, they are far from what investing experts do. In fact, most expert investors would use other indicators to make their decisions. One such indicator is the P/E ratio or price-to-earnings ratio (Frankel, 2023).

The P/E ratio divides the prices of the stock by the earnings per share. A high stock price, and low earnings would give a high ratio. A low price, and high earnings would give a low ratio. In general, good investors want to buy companies that are cheap, and are making a large profit. This is represented by a low P/E ratio (Frankel, 2023).

The logic behind this is simple. A company that is making a large profit but is cheap is undervalued. Eventually, investors will realize that this company is profiting and invest in it hoping to get some of that profit. This demand for stock will drive up the stock price. So, it is better to invest in a company that’s making a large profit when the stock is cheap.

The market data from India includes indicators such as P/E ratio, P/B ratio, and dividend yield in addition to stock prices and volume (Sahoo, 2023). Using these indicators and the machine learning algorithms that showed the most promise in predicting stock prices it would be possible to create a more accurate model. The gap that the other attempts had is that their data did not include some of the more used indicators.

# Proposed Project

The traditional investment process is about 80%-90% a waste of time. You spend 80%-90% of your time looking at stocks that aren’t worth an investment. If you could use machine learning to filter out those stocks that aren’t worth consideration and be able to just focus on those that are, it would save a lot of time.

This project would benefit any investors who spend a lot of time deciding on which stock to buy. Many investing strategies involve using key attributes like P/E ratio. This project should reveal which key attributes are the best for predicting stock prices and help investors who are uncertain about which strategy is best.

### Key Performance Indicators

The goal is to predict a future value based on previous values. This makes this a regression problem. The success or failure of a regression model depends on how many errors it makes and how big the errors are. To achieve the business objectives of increasing efficiency, reducing risk, and maximizing profit a model is needed that makes fewer errors than the alternative human centric approach. Therefore, the key performance indicators will all be measurements of the errors the models make. The goal is to create a model that makes better decisions, in other words fewer errors, than a human who is a professional investor.

For this assignment four machine algorithms will be tried. They are ARIMA, long short-term memory (LSTM), linear regression, and K-Nearest Neighbors. ARIMA, LSTM and linear regression use Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Squared Error (MSE) to measure a model’s performance. K-Nearest Neighbors usually uses accuracy, precision, recall, and a lift chart to measure accuracy. Therefore, this project will have the following key performance indicators.

Accuracy is an indicator of how often a model is correct overall. Precision is how often a model is correct when predicting a target class. Recall is how often a model can find all objects of a target class. The target class in this case is the correct stock price which is not a class. Unfortunately, when it comes to K-Nearest Neighbors the traditional performance indicators might not be adequate because we are not using it on a classification problem. The solution in a similar project was to measure accuracy by comparing a plot of actual stock price data, with a plot of stock prices predicted by K-Nearest Neighbors (Paramartha, 2022).

Means Absolute Error (MAE) is a powerful metric to evaluate the accuracy of regression models. It measures the average absolute difference between the predicted values and the actual target values. Since it doesn’t square the error that means it give equal weight to large errors and small errors. The value of 0 indicates no error, and a value of 1.0 indicates the biggest difference between predicted and actual value. This metric represents a linear increase in the absolute error value (Brownlee, 2021).

Mean Square Error (MSE) is another popular metric for evaluating the accuracy of regression models. It is calculated as the average of the squared difference between predicted and expected target values. The squaring causes large errors to be magnified which results in a curve and not a line. A value of 0 indicates no error, and a value of 1.0 indicates the maximum error. This metric is useful for measure how big of an error the model makes (Brownlee, 2021).

Root Mean Square Error (RMSE) is very similar to the Mean Squared Error (MSE). The big difference being that it is the root of the MSE. This has the effect of reversing the squaring of the error but keeps the value positive. This is used when you want to magnify large errors, but at the same time keep the value of the error to be in the units and not units squared. A value of 0 indicates no error, and a value of 1.0 indicates the maximum error (Brownlee, 2021).

Mean Absolute Percentage Error (MAPE) is one of the most common metrics for predicting regression model accuracy. This error is calculated like MAE with the big difference being that it displays a percentage. The advantage of this is that it is a more accurate way of measuring how large of an error was made. 0% means no error, and theoretically the error value can go up to infinity (Brownlee, 2021).

### Project Insights of your Data Analysis

Based on similar projects there are two possible scenarios. The first is that one or more algorithms can successfully predict stock prices. The second is that all the algorithms fail to successfully predict stock prices. There are examples for each algorithm of success, and failure. This suggests that the algorithms are good, but that the issue might be the data. In all the examples researched the data was limited to stock prices and volume.

Compared to the other examples there are three additional attributes included in this analysis. There is the P/E ratio, the P/B ratio, and the dividend yield. Conventional investment wisdom suggests these attributes play a role in whether the price increases, and how much the price increases. Based on that it is more likely that at least one model will be successful.

However, it is important to remember that a stock price reflects demand for a stock. There is no way to objectively measure demand. It is only possible to measure conditions that would in a normal economic situation increase demand, and as a result increase stock price. Not all the attributes can be accounted for, and there might be an abnormal economic situation in the data. The quality of the data is the most important factor in this project.

# Project Milestones

This project will use the CRISP-DM model and the milestones will be set as per that standard. The first milestone will be business understanding of the problem. This will be followed by data understanding. Then data preparation will be done. Then the machine models will be created, trained, and tested. The results will be evaluated, and improvements made. Based on the improvements the previous milestones might be repeated. Finally, the model will be deployed (IBM, 2021).

# Completion History

|  |  |
| --- | --- |
| **Week 1** | 1. Searched and chose several data sets |
| **Week 2** | 1. Developed the project scope |
| **Week 3** | 1. Reviewed the data sets 2. Chose the data set to be used for the project |
| **Week 4** | 1. Chose the approach for analyzing the data 2. Completed the business understanding 3. Created a data set description 4. Completed the appropriate portions in the project document |
| **Week 5** | 1. Completed a presentation on the project |
| **Week 6** | 1. Wrote a Python script to prepare, clean, and divide the data 2. Prepared, cleaned, and divided the data |
| **Week 7** | 1. Created visualizations for the data |
| **Week 8** | 1. Presented the project status 2. Updated the project |
| **Week 9** | 1. Created the models |
| **Week 10** | 1. Tested and updated the models |
| **Week 11** | 1. Presented results |

# Lessons Learned

|  |  |
| --- | --- |
| **Week 1** | That there is a lot of stock market data out there, but very little of it has more attributes than price and volume. |
| **Week 2** | The importance of project scope. How to determine project scope. |
| **Week 3** | Sometimes you must change the project scope to fit your data if you can’t find data to fit the scope. |
| **Week 4** | The different approach’s others have used for this problem. The different machine learning algorithms that can be used for such a problem. The results of those approaches and algorithms. How to measure success when trying to predict values. |
| **Week 5** | How to present a data analysis project appropriately to a target audience. |
| **Week 6** | The different approaches and methods of cleaning data in Python. How to create new features, remove unnecessary features, drop rows, and treat time series data for outliers. |
| **Week 7** | Learned how to visualize time series data. |
| **Week 8** | Learned alternative ways to set KPI boundaries and updated the KPIs. |
| **Week 9** | Learned how to create SARIMAX, LSTM, and Linear Regression models in Python. |
| **Week 10** | Learned how to tune models and measure their performance. |
| **Week 11** | The models the performed best and why they performed best. |
| **Week 12** | The intricacies of time series data analysis. The different models that can be used for such a problem and which models are best. How to create, test, and measure model performance. The history of model development, and how models are improved overtime. |

# Data Set Description

The first data set is the U.S. stock market data. It consists of one table for every company on all the U.S. exchanges. Each table has the following variables: date, volume, high, low, and closing prices. This makes the total number of features to be 5. The data is updated weekly. The range of the dates is different for each company, but there is some overlap between many of the companies. The number of records varies for each table. Most of the data is good, but there are some tables that have missing values. There is 10.23 GB of financial data that is split into too many files to do a row count quickly.

The second data set is the stock market index data from India (1990 – 2022). It consists of two tables for each market in India. One of the tables contains the following variables; date, open, high, low, and close. The other table contains the following variables; date, P/E, P/B, and Div Yield %. This makes the total number of features to be 7. The number of records varies between tables. Most of the data is good, but there are some tables that have missing values. The total row count is 81,070 when the two files are combined into one table.

The third data set is U.S. stock data set that includes only 514 stocks but has 1,298 features. Since there are 1,298 variables it is impossible to list them all in this document. It has the common variables of date, open, high, low, close, and volume. Plus, many more variables for many of the derived indicators that are used to predict stock market prices. It is approximately 3,000 records per stock, which makes the total number of records to be around 1,542,000.

All the data sets have many total records with very few missing values. They all have several indicators that can be used to train a model to predict price. The data of 514 U.S. companies has the most indicators, followed by data from India. Of all the data sets considered these three data sets had the highest number of records, most predictors, and fewest missing values. These three data sets also had the highest number of features.

The two tables for each market from the India would require an inner join by date to get the most value out of that data. Although it is possible to inner join all the data by date it might be counterproductive. First there is the fact that the dates don’t match on all records. Second, not all the variables match between three data sources. This would increase the missing values and result in less useful data. Choosing one of the data sources is the best approach.

Using transfer learning or bagging might be a better approach to joining all the data files into one large data file. Transfer learning is the act of training the same model with multiple data sets. Bagging is the act of training several models on several data sets and then averaging out their results. Since combining all the data will result in low quality, and less usable data these approaches are better.

The best data is the data of 514 U.S. stocks because it has the most variables. Since this data set has adequate records, it will be the one that will be used for this project with the others acting as backups. Also, since all the data sets have adequate records there is enough data to create training, validation, and test sets. Another reason to not use real life sampling and instead use test data from the data sets is that it is difficult to find real life samples that contain some of the variables. Getting real life samples of stock data is easy enough, but some of the variables will have to be calculated from the gathered samples which can be a time-consuming process.

### High-Level Data Diagram

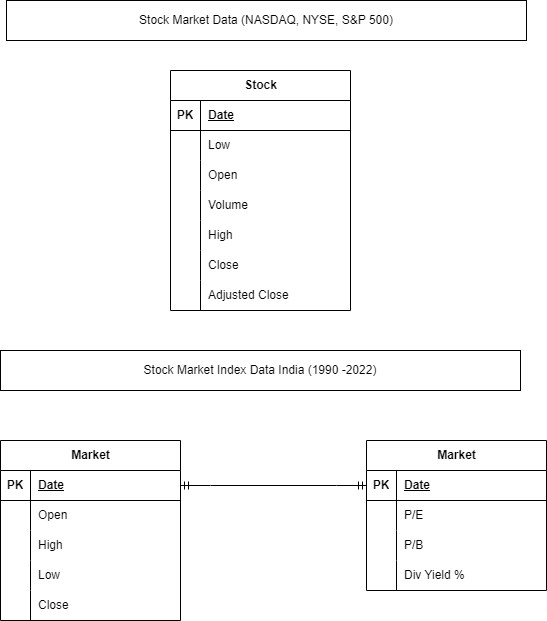
**

Figure 1: High Level Data Diagram

This diagram depicts the two data sets and their structure. Each data set consists of multiple tables, one for each stock in the U.S. data, and two for each market in the India data. The tables are structured as shown in the diagram. The date is the unique identifier and for the India data it can be used to join the two tables for each market. Since the data of the 514 stocks had 1,298 features it was not included in this diagram.

### Data Definition/Data Profile

The variables in the U.S. data include date, volume, high, low, and close. The variables in the India data include date, open, high, low, close, P/E, P/B, and Div Yield %. The data of 514 U.S. stocks includes the common variable of data, open, high, low, close, volume, and 1,292 other variables. The variables that all three data sets share are date, high, low, and close.

Date is the date of observation and is in the date type. High is the high price on a particular day and is numeric. Low is the low price on a particular day and is numeric. Close is the final price on a particular day and is numeric. Open is the opening price on a particular day and is numeric.

Volume is the number of stocks and is numeric. P/E is the price-to-earnings ratio and is numeric. P/B is the price-to-book ratio and is numeric. Div Yield % is the percent of dividends and is numeric. All numeric values other than volume are floating point types.  
 There are some quality issues with some of the data. Some records have missing values. Also, for many different stocks and indexes the dates don’t perfectly match up. Combining the three sets of data is going problematic as well. There are many columns that aren’t present in all three data sets.

The U.S. data containing 514 stocks has 1,298 columns which might have too many features. Also, several of the indicators are listed more than once for each record. For example, for one record the Bollinger Bands DEMA indicator has 30 different variables. For these types of variables, they are calculated using the previous record ranging from 3 to 200. Each column uses a different amount of previous data to calculate its value.

# Data Preparation/Cleansing/Transformation

### Data Preparation

The data that was chosen was the 514 U.S. stocks data set. This data consists of over a thousand csv files divided by temporal time frame, and stock index. There is a total of 514 stock indexes, and three temporal time frames. The temporal time frames include daily values, weekly values, and monthly values. This means that for each stock there are three files which brings the total number of files to 1,542. Each file may have several hundred to several thousand records depending on the file type. The files with daily values have the most records, and the files with monthly values have the fewest.

Not all files are required for our analysis. Therefore, the first step was to choose which files would be selected. The process was partially random, and partially systematic. First the files that did not have enough records to be used in the model were eliminated. This included all the files with monthly and weekly values. That left only 514 files with the daily values.

The next step was to randomly select 100 of the 514 files and after cleaning them to divide them into training, validation, and test sets. 100 files resulted in 488,925 records which was more than enough for the three data sets. There were 60 files in the training set, 20 in the validation set, and 20 in the test set. Since transfer learning or bagging will be used to train the models, the files were not joined. This was mainly done because the stock price varies from stock index to stock index and by joining the files outliers would be created. Then the treatment of those outliers could result in entire files worth of data losing their original values if one of the files has stock prices that are several orders of magnitude higher or lower than the rest. This would defeat the reasons for joining the files in the first place.

### Data Cleansing

The tools used for data cleaning were Python and the Pandas library. Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum in 1991. Python has multiple applications including data analysis (OpenAI, 2024). Since python is iterative it works well with iterative processes like the CRISP-DM process. Another advantage of using Python is that you only must write the script once, and even if the data changes the cleaning can easily be repeated just by executing the script.

The Pandas library is an open-source Python library that provides high-performance data manipulation and analysis tools. It is built on top of NumPy (Numerical Python) and is widely used for working with structured data (OpenAI, 2024). Since it is easy to convert csv files to pandas two-dimensional DataFrame it makes it ideal for working with csv files. It is also very easy to clean DataFrames and save the results in a csv file.

There are three parts to data cleaning: treating the data for missing values, dropping unnecessary variables, and treating outliers. First the data was treated for missing values. Some of the indicators in the data don’t start until row 201 because they need 200 rows of previous data to calculate. This makes the first 200 rows of every file unusable for the models. Using the Pandas dropna() method the first 200 rows of every file was easily dropped with one line of code. When this process was complete 59,400 records were dropped resulting in 488,925 total records remaining.

The next step was to drop unnecessary variables. There is some redundancy in the data because there are several indicators that have many variables. Each indicator is calculated using a certain number of previous records ranging from 3 previous records to 100 previous records. Each variable represents the number of records used to calculate that indicator. Since the variables that are calculated using the most records are the most accurate, they were kept. All other duplicates were dropped because they were redundant.

The variables open, high, and low are not indicators so they were also dropped. These three variables could be alternative target variables, but in general are not used in stock price prediction. Before they were dropped, they were used to calculate rolling averages which were added to the cleaned data frame (See Data Transformation for more details).

Since there are 1,298 variables to consider rather than dropping each variable one at a time, a list of all the variables that would be kept was made. Then a query, containing only the variables in that list, of the uncleaned data frame was taken. Then this original data frame was replaced with this query. This was easier than dropping several hundred variables. The results were that 1220 variables were dropped, and 78 were kept.  Since this is a time series data only local outlier were treated. A window was set to 10 values, and if there was an outlier within 10 values that was more or less than 30% of the local values it was replaced with the closest non-outlier value. Since time series data changes over time, the slope of the change could result in acceptable values being treated as outliers if the slope drastically changes in a short interval. This could cause the treatment of outliers to average out the results of large chunks of non-outlier values if done incorrectly. Therefore, it is important to keep the window small, and have the threshold slightly higher than the maximum 20% increase that is typically seen on daily stock market increases.

### Data Transformation

Of the 78 variables which were kept, 74 were present in the original data. Four new variables were created. These variables are the rolling averages for the open, high, low, and closing prices. Previous stock prices can be volatile and are generally thought of as bad predictors of future stock prices and are therefore rarely used. So, variables other than the target variable that contained base stock prices were dropped.

There are many kinds of averages that are used in stock price predictions. Two examples of averages that were included in the data set are Double Exponential Moving Average (DEMA) and Simple Moving Average (SMA). This data set had all the moving averages that are used in stock price predictions. However, there were no rolling averages in the data. A moving average is calculated using the average for a specific period. For example, in this data set the moving averages use the 100 previous records to calculate the average. For every new record, the 100 previous records changes by one record. A rolling average uses all the available data up to that record to calculate the average (IBM, 2024).

Since no rolling averages were included, 4 simple rolling averages were created. This was done by using the rolling method to compute the averages and setting the window length to the length of the DataFrame to include all possible values. The minimum period was set to 1 to include the first line even though it has no previous values to create an average from. The results are a rolling average for the opening price, a rolling average for the high price, a rolling average for the low price, and a rolling average for the closing price.

### Data Analysis

Since Python is going to be used for the entire project the tools used for visualization and predictive modeling are all Python libraries. Since the CRISP-DM process is iterative it works well with iterative tools such as Python. The data is going to be put into Pandas DataFrames because there are many libraries that work well with the Pandas DataFrames (Pandas, 2024). The following were chosen for both their usefulness on the current project and their compatibility with Pandas DataFrames. For visualization Matplotlib, Seaborn, and Plotly will be used. For predictive modeling Scikit-learn, Statsmodels, and Tensor flow will be used.

Matplotlib is a general python data visualization library that works with many types of data. Although the visualizations are simple, they are comprehensive and work well with Pandas. Seaborn is a library that is based on matplotlib but also has statistical estimation features and other additional features. Plotly allows for the creation of interactive plots in WebGL and D3 that are more visually appealing. These three were chosen because together they work well with Pandas and cover almost any visualization scenario that the analysis could require.

Predicting stock prices is a regression problem and not a classification problem. Therefore, the four algorithms that will be used in this analysis are algorithms that can handle regression problems with time series data. The four algorithms have a good track record of solving regression problems with time series data are linear regression, k-nearest neighbors, ARIMA, and long short-term memory.

Linear regression is a machine learning technique used to model the relationship between dependent variable and one or more independent variables. The relationship is a linear equation. ARIMA is a machine learning algorithm used to forecast time series data such as stock prices. Long short-term memory is a type of recurrent neural network designed to model sequential data and overcome the limitations of traditional recurrent neural networks in capturing long-term dependencies. Long short-term memory algorithms are particularly effective in tasks involving time series predictions. The k-nearest neighbors algorithm is a simple and effective supervised learning algorithm that can be used for regression tasks. Instead of making assumptions about the underlying data distribution it instead makes predictions based on similarity of new data to know data points in the training data set (OpenAI, 2024).

The Scikit-learn library supports linear regression and K-Nearest Neighbors models. Statsmodels supports ARIMA models, and Tensorflow supports long short-term memory models. All three libraries work well with Pandas data frames.

Three of the algorithms will use means absolute error (MAE), means square error (MSE), root means square error (RMSE), and mean absolute percentage error (MAPE). These can be visualized using a scatter plot, box plot, or a heatmap. K-nearest neighbors measures how close the test values are to the training values. It is best visualized with a lift chart.

For each of the error metrics there is an acceptable value. The values in the data set range from 0.2 to 1,147.56. Since the range of the target variable is quite broad, it means that the error metrics must be less than 5% of the max value or less than 57.38 (OpenAI, 2024). For MAE, MSE, RMSE, and MAPE the value should be less than 10% of the maximum value (OpenAI, 2024). That means that the target value should be less than 114.76. Finally, in the lift chart the predicted values and actual values should be close together.

# Data Visualization

### Data Visualization 1 – Line Plots

|  |  |
| --- | --- |
| A graph with blue lines  Description automatically generated | A graph with blue line  Description automatically generated |
| Figure 1 | Figure 2 |
| A graph with a line graph  Description automatically generated | A graph with blue lines  Description automatically generated |
| Figure 3 | Figure 4 |

The first visualization depicts 4 line graphs that depict 4 variables from the stock of Autodesk inc. These visualizations show how these variables have changed over time. The unit of measurement for the x axis is trading day converted into an index that represents each record. These were converted into indexes for the following two reasons. First, the dates are trading days which exclude weekends and some holidays, so it is an interrupted time series. Second the actual date is not important, but the order is important, for example the 10th trading day. The y axis is the value of the variable on a given trading day.

Figure 1 shows the “close” variable which represents the closing price. This is the target variable, and it is important to see how it changes over time. This stock was chosen for visualization because it has both the typical gradual increase and sudden unexpected changes portions to the closing price in its line graph. Figure 2 is the double exponential moving average (DEMA) indicator over time. This indicator was chosen for visualization because it is a moving average indicator. This means that it will closely resemble the closing price, and as a result the line plots will be almost identical.

Figure 3 shows the parabolic SAR extended indicator (SAREXT). This indicator is used to track changes in the price by using extreme highs and lows in a period. It tracks trends, whether a price is going up or down, and not the actual price. Buy trends are indicated when the value is above zero, and sell trends are indicated when the value is below zero with the values representing their magnitude (Open AI, 2024). This variable was chosen because it is one of the trend indicators, and it would not resemble the closing price.

Figure 4 shows the Hilbert transform dominant cycle period indicator (HTDC period). This indicator is used to identify the dominant cycles in time series data. The y value represents the dominant cycles and their magnitudes. Each peak represents a dominant frequency (change in direction), and the distance between peaks represents the length of the dominant frequency (Open AI, 2024). This variable was chosen because it is used to analyze cycles and not stock prices. In other words, it is likely to be a poor predictor of stock prices.

These four variables do a good job of displaying the range relationship between the closing price and each indicator. The closest are the moving averages with DEMA being almost an exact copy of closing prices. Then in the SAR extended indicator the prices are lost, but the changes in price are still retained. Finally, the HTDC period has no visible similarities to the closing price. All other indicators fall somewhere in this range.

In a regression analysis of time series data understanding and measuring how each independent variable influences the dependent variable is important. However, since some of the independent variables are calculated from the dependent variable a line chart cannot tell if an independent variable is a good predictor. It can only be used to show that there is some sort of relationship between the two variables.   
 The results of the line charts are as expected and are not enough to influence the scope of the project. The fact that some variables are more closely related to the closing price than others can also be ascertained by reading the description of how those variables are calculated. A line chart is just visual proof of a known fact. Also, a line chart isn’t enough information to make any changes to the scope.

### Data Visualization 2 – Scatter Plots

|  |  |
| --- | --- |
| A graph of a line  Description automatically generated with medium confidence | A graph with blue dots  Description automatically generated |
| Figure 5 | Figure 6 |
| A graph of a scatter plot  Description automatically generated | A graph with blue dots  Description automatically generated |
| Figure 7 | Figure 8 |

Once a relationship between two variables has been established, a scatter plot can be used to visualize the strength of that relationship. A scatter plot visualizes the relationship between a dependent variable and one independent variable. Ideally a scatterplot should show both a positive correlation and a strong relationship. A positive correlation is when the points tend to slope upwards from left to right, and a strong relationship is when they are clustered around a trend line. Positive correlation means that when one variable goes up so does the other. A strong relationship means that if one variable goes up by a certain amount, calculating the change in the other variable can be done consistently.

Figure 5 shows the relationship between the closing price and the Bollinger bands (BBANDS) indicator. There is a positive correlation here and a somewhat strong relationship. Figure 6 shows the relationship between closing price and double exponential moving average (DEMA). There is also a positive correlation in this plot, and a much stronger relationship.

Figure 7 shows the relationship between the closing price and the parabolic SAR extended indicator (SAREXT). There is a very strong relationship between these two variables. However, the variable is split between values that are positively correlated, and values that are negatively correlated. This makes sense if you consider that this variable is used to track changes in the prices. All the values in the closing price would be positive, but if the price drops some of the values in the SAREXT variable would be negative to indicate that the price dropped. The magnitude would not change, and therefore the strength of the relationship would not change.

Figure 8 shows the relationship between the closing prices and Hilbert transform dominant cycle period indicator (HTDC period). There is no correlation between these two variables as the points are scattered all over the plot. Since the points are scattered, it also means that the relationship is very weak. This makes sense if you consider that this variable is used to measure cycles and not trends or stock prices. Although only one visualization is displayed here, this is a common pattern for all 5 cycle indicators in the data set.

The indicators that are calculated using the stock price are usually the ones that have the strongest relationship and most positive correlation. There are exceptions such as the BBANDS indicator which has a weaker relationship. Some variables that are used to measure trends also have a very strong relationship with the closing price, such as SAREXT. However, in this case the correlation is not always positive.

The five cycle indicator variables included in the data have no correlation and a weak relationship with the dependent variable. This makes sense since their purpose is to measure cycles. A cycle is a recurring pattern that represents investor behavior. These patterns are influenced by a variety of economic factors and market dynamics. Although the actual pattern, the economic factors, and the market dynamics might be useful in stock prediction, these variables don’t depict those things. Instead, they depict the start and end of such a pattern.

At this point in the analysis most variables have either adequate correlation, a strong relationship, or both to be useful for the models. The exception here are the cycle indicators. These include Hilbert transform dominant cycle period, Hilbert transform dominant cycle phase, Hilbert transform phasor components, Hilbert transform SineWave, and Hilbert transform trend vs cycle mode. These don’t appear to have any relationship to the dependent variable and should be dropped.

### Data Visualization 3 – Autocorrelation Plots

|  |  |
| --- | --- |
| A blue graph with white text  Description automatically generated | A blue graph with white lines |
| Figure 9 | Figure 10 |
| A graph with a line  Description automatically generated | A blue triangle with white background  Description automatically generated |
| Figure 11 | Figure 12 |

The final visualization is 4 autocorrelation plots of the four variables that were previously analyzed. Autocorrelation is the measurement of the degree of similarity between a time series and a lagged version of itself over successive time intervals (Open AI, 2024). In other words, how well do past values correlate with future values. On the x-axis is the time lag, and on the y-axis is the correlation. A correlation of 1 indicates perfect correlation and is usually the value for a time lag of 0. A -1 indicates an inverse correlation and a 0 indicates no correlation.

This is important because when it comes to time series data, the future values are often predicted using past values. An autocorrelation plot shows how similar past values are to future values for a given variable. Although it doesn’t directly translate into predictive power, it can be used to increase the predictive power of time series models. This is often used for fine tuning time series models like ARIMA that give the option to specify how many past records to include (Open AI, 2024).

Figure 9 shows the autocorrelation plot for the closing price. At about 2000 records the correlation starts to drop below 0.2. This means that when predicting future closing prices using past closing prices, including more than 2,000 past closing prices would reduce the predictive power of the model considerably. Figure 10 shows a very similar graph for the DEMA indicator.

Figure 11 shows poor autocorrelation for the SAREXT indicator for most past values. This means that for this variable it is best to use only the most recent past values to predict future values for best results. This is most likely because this variable shows either the magnitude or the inverse magnitude depending on which way the trend is heading. According to Figure 1/Figure 3 the direction of increase or decrease changes often. This means that the value of this variable changes from positive to negative to positive very often making it almost impossible to use past values to predict future ones. This variable might be good for models where relationships are important, it is not good for models where past values are considered.

Figure 12 shows the autocorrelation for the HTDC period indicator. There is acceptable autocorrelation up to a little over 4,000 records for this variable. In spite its other short comings this variable falls into a pattern where the past values can be used to predict future values reliably. Although this variable might be poor for models where relationships are important, it is good for models where the past values are considered.

In some time series models, such as ARIMA, the number of past records included needs to be specified. For other models, such as linear regression, it is not important. These models make their predictions using different methods, and as a result require different considerations. For those models that require the number of past records to be specified autocorrelation plots are very useful in fine tuning those models.

In general, most of the variables that are derived from the closing price have very similar autocorrelation plots. In most cases there are several thousand past records that can be used to predict future results. In other cases, there are several hundred. In a few cases only the most recent previous values can be used to predict future values.

These findings change the scope slightly because the data set that is optimal for linear regression might not be ideal for models like ARIMA. For example, data that is ideal for linear regression would contain variables with strong relationships that are positively correlated. Data that is ideal for ARIMA would contain variables with high autocorrelation. To fine tune both models it might be better to create two data sets fine tuned for each model rather than use the same data set for both models.

### Proposed Visualizations

One additional visualization that might be useful for this analysis would be a histogram. A histogram is a graphical representation of the distribution of numerical data. It consists of a series of contiguous bars. Each bar represents the range of data values, and the height corresponds to the number of data points. Histograms are used to understand how data is distributed (Open AI, 2024)  
 Since the application of a histogram in time series data is limited to variables such as SAREXT, it wasn’t used. However, it would be useful in analyzing those variables, and the distribution of upward trends and downward trends in a time series data. This could help determine if those variables are skewed or help detect outlier.

Another way to visualize correlation between two variables is with a heat map. A heat map is graphical representation of data where values are represented as colors on a two-dimensional grid. It is good at visualizing the magnitude of relationship between two variables. Each cell in a heatmap is colored according to the value of the variable it represents, and each color is from a gradient palette where different hues represent ranges of values. A heatmap allows patterns and trends to be visually discerned based on color intensity (Open AI, 2024).

In correlation analysis high correlation is represented by more intense colors on a heat map. Although a heat map does the same job as a scatter plot it is easier to spot high correlation on a heat map. Since correlation is one of the most important factors in this analysis a heat map would be a very useful visualization. A scatter plot also shows correlation, the strength of the relationship, and any outliers. Since the scatter plot was able to show more useful information than a heat map, the heat map was not chosen.

# Predictive Models

### Predictive Model 1: Linear Regression

Linear regression can be represented by the following equation.

* is the dependent variable (what you want to predict).
* is the independent variable (the predictor).
* is the y-intercept (the value of y when = 0.
* is the slope of the line or the coefficient for each independent variable. (how much y changes for a one-unit change in x).
* is the error term (the difference between observed and predicted values).

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It tries to find the best fitting line that predicts the dependent variable using independent variables (OpenAI, 2024). The biggest advantages of linear regression are its simplicity, speed, and low computational resource requirement. The problem with linear regression is that it assumes that the relationship between the independent and dependent variable is linear. This means that in general it has more trouble capturing more complex relationships.

A linear regression model that uses stochastic gradient descent was created. This is a type of linear regression model that updates the model weights incrementally which is especially beneficial for large data sets (Scikit, 2024). Since this model requires little processing time there was plenty of time to adjust the parameters. All 4 loss functions were tried. Those loss functions are squared error, Huber, epsilon insensitive, and squared epsilon insensitive. Additionally, the penalty, max iteration, and learning rate were adjusted to several settings.

Transfer learning was used to train a single model on multiple data sets. This was done by using the ‘partial\_fit’ method that allows one model to be partially fit with multiple sources of data. When using this method, it is best to use data that has zero unit variance (Scikit, 2024). This means that this method is very sensitive to scale, and that all the data should be on the same scale. Each data source used is a different stock index that often has a different scale, and this model cannot address this problem.

No matter what model settings were tried the results were not good (See Table 1). None of the results were within the acceptable range for every metric. Although the epsilon insensitive loss function performed the best, its MAE was which is many times higher than the max acceptable value of 57.38.

Table 1: Sample Results for Linear Regression (AAL Stock)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Loss Function | MSE | MAE | RMSE | MAPE |
| Squared error |  |  |  |  |
| Huber |  |  |  |  |
| Epsilon Insensitive |  |  |  |  |
| Squared Epsilon Insensitive |  |  |  |  |

There are several reasons why the error values could be so high. First is that the range of the target variable is very high. However, when calculating the acceptable error values this was considered so this is probably not the reason. Another reason is high variability of data. Besides the variable ‘volume’ most of the other variables do not have high variability in data. However, since multiple data files and transfer learning was used all the variables could have high variability if those data files were on very different scales which is very likely. This is a strong explanation that is supported by limitation of the linear regression model in handling different scales of data.

Another reason could be the presence of outliers. Since this is time series data getting rid of outliers is a bit trickier than in other types of data. The probability that outliers were overlooked is high and could contribute to such error values but there are better explanations. For example, the model couldn’t fit properly because the data was not linear. This is very likely the case because stock data has many ups, and many downs. A linear algorithm will make a lot of errors if it tries to draw a straight line through data that fluctuates often. Therefore, the best explanation for these results is that the data is not linear and has high variability.

All of this means that linear regression is an algorithm that is poorly suited for predicting stock prices. It doesn’t handle non-linear data very well. It doesn’t handle variability very well. It doesn’t handle time series data very well. Since stock data is non-linear time series data with high variability this algorithm made a lot of errors.

### Predictive Model 2: SARIMAX

SARIMAX can be represented by the following equation.

* is the value of time series at time t.
* is the mean of the series.
* is the coefficients for the auto regressive part.
* is the coefficients for the moving averages part.
* is the seasonal lags for the time series.
* is the season’s auto regressive coefficient.
* is the white noise error term at time t.

SARIMAX is a type of ARIMA model that supports seasonal behaviors and exogenous variables. ARIMA stands for AutoRegressive Integrated Moving Average. SARIMAX stands for Seasonal AutoRegressive Integrated Moving Average with eXogenous factors. Both are popular statistics models used in time series forecasting (OpenAI, 2024). SARIMAX was chosen instead of the standard ARIMA models because of its support for exogenous variables. Standard AutoRegressive models use auto regression analysis on the target using past target variables to make predictions. SARIMAX includes the use of other variables so it in fact uses both regression and auto regression.

The disadvantages of SARIMAX are that it is very complex with many seasonal and exogenous terms that make it hard to tune. Tuning this algorithm is very challenging and requires experimentation. This complexity also increases the computational time significantly. This makes tunning even more difficult since one model could take up to an hour or more to train. Also, it does not support transfer learning and as a result bagging had to be used. This meant that using bagging for 60 data sets it could take as much as 60 hours to train.

With those considerations the first thing that had to change was that the training, validation, and test sets had to be reduced. This was changed to 20 in the training, 5 in validation, and 5 in the test sets to reduce the computation time. Even with those changes the computational time was significant and not all parameter combinations were tried. For the order the best results occurred when p, d, and q were all set to 1. For the seasonal order the best results occurred when p, d, and q were set to 1. The parameter s was set to 5 to represent the 5-day trading cycle which had the best results. However, it worked well with other settings such as 12.

Although the results of these settings were adequate, there is still a lot of fine tunning that could take place. When using bagging in regression models it is common to train, validate, and test several base models independently. Then their results are average out to get the overall result (OpenAI, 2024). After bagging the results were acceptable and are shown in table 2 (See Table 2).

Table 2: SARIMAX Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of Model | MSE | MAE | RMSE | MAPE |
| Bagging (Season Order (1,1,1,12) | 1.1967 | 0.0044 | 0.0861 | 0.0640 |
| Bagging ( Season Order 1,1,1, 5) | 0.0007 | 0.007 | 0.0076 | 0.037 |

All the error values are far below the acceptable error range for all the error metrics. This is not surprising because this algorithm was designed to handle this kind of data. It is designed to work with time series data, where it changes based on time units and time unit cycles. It considers external factors as well. It uses both regression and autoregression when building models.

Although several models were tried, the two of best models are displayed in Table 2. The only thing that changed between these models was the seasonal cycle. The model where it reflects the 5-trading day cycle of the stock market performed the best. This demonstrates SARIMAX’s ability to recognize time cycle patterns in time series data.

One thing that is interesting about the results is that like linear regression, SARIMAX also assumes that the relationship between time series values and the predictors is linear. Regardless of having similar assumptions one model performed exceptionally better than the other. One possible explanation could be the seasonal component. Since SARIMAX breaks down the time series data into cycles it is better at handling fluctuations. It also assumes that the data could be made stationary. This means that the mean, and variance don’t change over time.

The amount of time it takes to train a model is very high. Also, there is no support for transfer learning so the best that can be achieved is to use insights from one data set to tune the parameters on a SARIMAX model on another data set. That means that creating and deploying the best SARIMAX model is a very time consuming and difficult process.

### Predictive Model 3: LSTM

The LSTM is a recurrent neural network that can be represented with the following equations.

1. Forget Gate

* is the forget get at time t.
* is the Sigmoid activation function.
* is the weight matrix for the forget get.
* is the previous hidden state.
* is the current input.
* is the bias for the forget gate.

1. Input Gate

* is the input gate at time t.
* is the weight matrix for the input gate.
* is the bias for the input gate.

1. Candidate Cell State

* is the candidate cell state at time t.
* is the weight matrix for the candidate cell state.
* is the bias for the candidate cell state.

1. Cell State Update

* is the cell state at time t.
* is the previous cell state.

1. Output Gate

* is the output gate at time t.
* is the weight matrix for the output gate.
* is the bias for the output gate.

1. Hidden State Update

* is the hidden state at time t.

Unlike the previous two models the final model is a recurrent neural network that is designed to learn from sequential data. Unlike other neural networks, LSTM can learn long-term dependencies in a sequence. LSTM uses memory cells that maintain information over long periods of time. This makes them good for handling time series data with long term dependencies. The data used includes variables that have significant autocorrelation for about 2000 variables into the past, which makes LSTM a very good model for capturing this dependency.

In an LSTM neural network, the cell state carries relevant information about the sequence. There are three gates that control the flow of information. The forget gate determines which information is discarded from the cell state. The input gate decides what new information gets stored in the cell state. The output gate controls what information is output from the cell state. The learning process follows a forget-input-update-output sequence. The hidden state is both the input and the output and is used to update the cell state (OpenAi, 2024).

Another advantage of LSTM is that is capable of learning from noisy data. This makes it a good fit for this data since it has a lot of irregular patterns. Also, it supports both transfer learning and bagging. With so many options, so many variables to tune, and the time it takes to train the model it was difficult to try every type of model.

LSTM is also a very complex model that requires more time to train. Although it is not as time intensive as SARIMAX, it is not far off. For those reasons the training, validation, and tests sets were reduced to 20, 5, and 5 respectively for most models. Despite these changes the processing time reduced the amount of fine tunning that could be done to optimize these models. Different numbers of hidden units were tried, and several activation functions were also tested. The best results were at 50 activation units with the hyperbolic tangent, or “tanh” activation function.

The results are a bit mixed, with some models meeting the acceptable values and other models not so much. Both bagging and transfer learning were tried with mixed results between the two. Although bagging generally performed better than transfer learning, it usually had higher MAPE values than transfer learning. Some of the results are displayed in Table 3.

Table 3: LSTM Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of Model | MSE | MAE | RMSE | MAPE |
| Bagging (Relu, Epoch 50, Units 50, Batch 32) | 0.0042 | 0.0392 | 0.0462 | 135.2359 |
| Bagging (Tanh, Epoch 50, Units 50, Batch 32) | 0.0032 | 0.0361 | 0.0430 | 119.6302 |
| Transfer Learning (Tanh, Epoch 50, Units 50, Batch 32) after 20 trainings | 1761.754 | 34.703 | 41.0242 | 71.306 |
| Transfer Learning (Tanh, Epoch 50, Units 50, Batch 32) after 80 trainings | 5462.556 | 62.34 | 69.977 | 74.6485 |

Although the bagging models had similar error rates compared to SARIMAX, the MAPE values were substantially higher. This means that when those models did make a mistake, they tended to make a bigger mistake than SARIMAX. The difference is not very significant, and it could be the result of non-optimal tunning of the LSTM model.

All the transfer learning models show a decreasing validation loss suggesting that the models generalized to the validation set. However, when tested against new data the models made more errors. Additional training did not necessarily improve the model. In fact, the models that were trained 20 times often performed better than those that were trained 80 times. All evidence suggests that transfer learning resulted in overtraining of the model.

Unlike the previous models, LSTM doesn’t make any assumptions about the data. Its performance can probably be attributed to the fact that it can work well with different types of data. Like SARIMAX it also uses exogenous variables as well as past values when training. The big difference is that it can use more past values than other models. The ability to work with all types of data and look farther into the past gives LSTM a big advantage over the other models. However, its ability to use more past data than other models might affect the model’s accuracy negatively if there are variables with poor autocorrelation. One thing that was not tried due to time constraints was the removal of all variables with poor autocorrelation to see if the accuracy of the model would improve.

Although the time required to train LSTM is high, the fact that it supports transfer learning means that what is learned can be saved. This means that the training process, although time-consuming, is not difficult because you don’t have to start over with each model. Although this could lead to overtraining, it also means that creating and deploying LSTM models is much easier than SARIMAX.

### 

### Predictive Model Review

Of the three models, linear regression failed to meet expectations. The problem is that linear regression is an algorithm that is poorly suited for predicting stock prices. It doesn’t handle non-linear data very well. It doesn’t handle variability very well. It doesn’t handle time series data very well. Since stock data is non-linear time series data with high variability this algorithm made a lot of errors.

SARIMAX made very few errors and the errors it made were less significant. However, the amount of time it takes to train a model is very high. Also, there is no support for transfer learning so the best that can be achieved is to use insights from one data set to tune the parameters on a SARIMAX model on another data set. That means that creating and deploying the best SARIMAX model is a very time consuming and difficult process.

Some LSTM models also made very few errors, but the errors they made were more significant. Other LSTM models showed signs of over training. It is important to note that there were a lot of tunning options that were not tried due to time constraints. Although the time required to train and tune LSTM is high, the fact that it supports transfer learning means that what is learned can be easily saved. This means that the training process, although time-consuming, is not difficult. Creating and deploying LSTM models is also less difficult.

The first consideration that needs to be made when proposing the best model is how much fine tunning was done. Due to the computational time of SARIMAX, and LSTM perhaps not enough time was spent fine tunning these models. The higher error scores on some LSTM tests could be attributed to poor tunning. At this point, most of the tests suggest that SARIMAX is the better performing model. LSTM could be a close second with more tunning and training. Linear regression is not adequate for this problem.

The second consideration that needs to be made is how easy are the models to train and deploy. Since linear regression has failed to meet the first criteria, only SARIMAX and LSTM will be compared. SARIMAX took much longer to train than LSTM. With some fine tunning LSTM error metrics could be improved, but there is no way to improve the training speed of SARIMAX that won’t also improve the speed of LSTM. Unlike SARIMAX, LSTM supports transfer learning, which means that previous training can be retained. This means that in the long run there is some possibility that LSTM models will meet or exceed SARIMAX’s performance. If immediate results are needed then SARIMAX is the best model, but looking long term LSTM has more pros than SARIMAX.

# Final Results

### Analysis Justification

Predicting stock prices using market data has many benefits in several fields. These fields include finance, data analysis, and statistics. In the field of finance doing so could improve return on investment by making data-driven decisions. It could also reduce risk by reducing the errors made. It can also reduce the time spent making investment decisions by automating some of the stock vetting processes. Saving time could also reduce the cost of investing as it would require less hours to analyze a stock and make a decision.

In the field of data analytics and statistics there are several benefits from such an analysis. The comparison of several algorithms that could be used to tackle this problem would help similar analysis in the future. There are many studies that are similar but focus on only one model. This analysis could aid future data analysts in choosing the best model for their data. It could also aid them in preparing data to work well with the models tested.

Time series analysis is one of the least common analyses done by data analysts (Kamenova, 2024). However, it could be argued that its applications make it one of the most important analyses that can be done. In addition to stock prices, time series analysis is very useful in analyzing time sensitive data such as economic data, demographic data, climate data, and astronomical data. For this study a subset of economic data was chosen. This study could very well be done with other kinds of time sensitive economic data.

The evolution of ARIMA into SARIMAX is an interesting historical phenomenon that has some serious implications for data analytics. First there was ARIMA, then it evolved into SARIMA, and finally into SARIMAX (Resende, 2024). At each stage of the evolution process another component was added. This evolution resulted in a very complex model designed to handle time series data. There are two interesting takeaways from this history. First, the model is getting more complex which indicates the complexity of the problem analyzing time series data presents. Second, is that perhaps the evolution of this model is not finished yet.

Another reason to do analysis such as these is to look for ways to improve existing models. SARIMAX may well be the best possible ARIMA type model, but taking what it does well and applying it to other models could improve other models. For example, LSTM is a much newer evolution of the recurrent neural network (RNN). It didn’t do as well on time series data as SARIMAX. This could be because it isn’t designed to deal well with stationary time series. Perhaps it is because it doesn’t use moving averages.

The comparison of the models in such an analysis could generate ideas on how to improve existing models or make new models by combining what previous models do well. Already this analysis has generated some ideas about making future models. Could you make a recurrent neural network that can keep track of long-term dependencies, account for stationary time series, and use moving averages in addition to auto regression to predict future values? Would such a model perform better than the current existing models? Would such a model be too complex? The results of this analysis are not limited to finding a way to predict stock prices and could result in better models if further study is done.

### Findings

ARIMA was first introduced by Box and Jenkins in the 1970s and is one of the most widely used models in time series forecasting. ARIMA has three components; autoregression, integrated, and moving averages. SARIMAX is the last stage of evolution of ARIMA as it also includes the seasonality, and exogenous variables (Resende, 2024). Of all the models tried, this was the only one that was designed to tackle this kind of problem. It is not surprising that it is also the model that performed the best.

The motivation behind LSTM is a bit different. LSTM is an improvement of the RNN. When classic RNN uses back-propagation the long-term gradients can disappear. This means that they tend to move to zero due to very small numbers creeping into computation, which causes this model to stop learning (Wikimedia, 2024). This means that traditional RNN tends to learn more from newer data, and “forget” older data. LSTM was designed for tasks such as natural language processing, where the grammatical dependencies need to be remembered. However, it’s “memory” function can be used to predict time series data as well. Its performance can be attributed to its ability to remember long-term dependencies. However, very likely the reason it didn’t do as well as SARIMAX is because it is being used for a task it wasn’t designed to handle.

Linear regression wasn’t designed for this kind of problem. The model is too simple to handle complex time series data. Unlike the other two models it only uses regression and does not include auto regression. It assumes that the relationship is linear, and it makes no attempts to make the time series data stationary. Therefore, it doesn’t handle variance very well which is bad because stock data tends to fluctuate often. Linear regression exceeded the acceptable error metric values.

SARIMAX had error values far below the acceptable limits. LSTM had some of the error values far below the acceptable limits. The best results for each model type are shown in table 4 (See Table 4). The criterion for choosing the best model is as follows. First the model had to have the lowest number of errors, which is measured by MAE. Second the size of the error was considered, which means that it had to have a low MAPE. Although both models made very few errors depicted by low MAE scores, any MAPE above 114.76 was deemed too high. Therefore, the best LSTM model didn’t quite meet the standards but got very close.

Table 4: Best SARIMAX vs Best LSTM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of Model | MSE | MAE | RMSE | MAPE |
| SARIMAX Bagging ( Season Order 1,1,1, 5) | 0.0007 | 0.007 | 0.0076 | 0.037 |
| LSTM  Bagging (Tanh, Epoch 50, Units 50, Batch 32) | 0.0032 | 0.0361 | 0.0430 | 119.6302 |

The best SARIMAX model was the one that first had the lowest MAE score, and second had the lowest MAPE score. This is the bagging model with a seasonal order of 1,1,1,5. In this model the seasonal cycle was set to 5, which represents the 5-trading day cycle of the stock market. Unlike the other models, SARIMAX is sensitive to seasonal cycles, and this could explain the results. MSE, MAE, RMSE, and MAPE are all very good and well below the limits. This model meets all KPIs.

There are some signs that the LSTM models suffered from overtraining. All the transfer learning models show a decreasing validation loss suggesting that the models generalized to the validation set. However, when tested against new data the models made more errors and larger errors. Additional training did not necessarily improve the model. In fact, the models that were trained 20 times often performed better than those that were trained 80 times. All evidence suggests that transfer learning resulted in overtraining of the model.

### Review of Success

Most of the models tried failed to meet the criteria outlined in the KPIs. However, both the SARIMAX models that were tried had error metrics far beneath the acceptable limits. The SARIMAX model that was tuned to the 5-trading day cycle performed the best. In this analysis the **data analytics problem that is being analyzed is using machine learning, with market data to predict stock prices.** Even though the error rates are low, the SARIMAX model meets the business objectives.

There are several general objectives in the investment industry. They are in no particular order; maximize profit, reduce risk, facilitate growth, capital appreciation, increase income, increase tax efficiency, liquidity, and speculation. To achieve success as an investor, organizations must meet the mentioned business objectives. There is another success criterion that isn’t mentioned but is a part of every business objective. That is to improve efficiency by reducing the time it takes to achieve those business objectives.

The SARIMAX model can be used to increase income. Since it can predict future stock prices with very few and very small errors it can be used to make data driven investment decisions. Compare this to a human centric approach, where attempts to predict the stock price are rarely made. Such attempts require complex computations that take a lot of time. Instead, most investors try to predict whether the stock will increase or decrease over a certain period. Since this model can exceed human capabilities in this area this objective is met.

The low error rates mean that the risk of making a mistake is less. When investing, humans often make mistakes. Often these mistakes are not optimal investments, but occasionally their mistakes lead to loss. One common strategy for dealing with risk that is used by human investors is to diversify. The goal of this strategy is that success in some investments exceeds the failures in others. The premise of this strategy is that bad investments are inevitable. This model significantly reduces the risk of a bad investment, by significantly reducing the error rate, and therefore meets this objective.

The time it takes for a human to compute this much data is significant. It is so significant that most investors don’t attempt to make such computations. The computations that are done, when done by humans, require more than one human to work multiple hours to achieve a similar result. In this analysis one analyst was able to do a job that would take many professionals weeks to complete. This reduces the cost of analyzing stock data, reduces the manpower required, and increases the speed. Therefore, the efficiency objective is met by this model.

Increased income, reduced risk, and increased efficiency all translate into increased profit. Therefore, the SARIMAX model meets this objective as well. The SARIMAX model has met the KPIs. The SARIMAX model has met 4 of the 9 business objectives of the investment industry. Therefore, the SARIMAX model has succeeded in predicting stock prices using historical data. Although not a perfect system, it exceeds human capability in all the areas mentioned above.

### Recommendations for Future Analysis

There are several improvements that should be implemented in future analysis. First, there could be an improvement in the data used. Second, the cleaned data could be better tailored for each specific algorithm. Third, the existing algorithms could be better tuned. Fourth, existing algorithms could be improved, or a new algorithm could be developed using what was learned from this analysis. Each of these improvements is described in greater detail below.

There were a few problems with the data used. First, it didn’t include some of the more common indicators such as P/E ratio. Second, some of the stock indexes included thousands of records, while others included less than a hundred. This means that often the time periods didn’t overlap. So, it is possible that some market conditions that were present in one data set were absent in another. Third, the data didn’t include other factors that could be affecting the whole economy and might influence stock prices. Data that contains these three things would result in better and more accurate models.

When it comes to cleaning the data better, the following should be tried. In two of the model’s autocorrelation played a role in the predictions. In both cases all the variables, including the ones that had poor autocorrelation, were used. For comparison, models that rely on autocorrelation should be tried only with variables that have good autocorrelation. Also, the SARIMAX model data could be cleaned in such a way that it always starts at the beginning of the seasonal cycle. That means that the first day should be a Monday for every stock index. These cleaning modifications should be tried, and the results should be compared to existing models.

Due to the complexity of some of the algorithms involved not enough time was spent tunning them. This complexity translated into more processing time, and in some cases 1 model could take several hours to process. This is particularly true with the LSTM model, which almost met the KPI. Some extra time tunning this model could increase its performance. For the SARIMAX model not all orders and seasonal orders were tried. Also, parameters such as “enforce stationarity” and “enforce invertibility” were only tried in the “true” setting. For the LSTM model parameter options such as the initializers, the bias, and the dropout were left in their default state for all the models due to time constraints.

Improving existing algorithms or creating new ones is not an easy process. However, to meet demand algorithms like ARIMA have been evolving over time. Since 1970 the model has had three stages. First there was ARIMA, then SARIMA, then SARIMAX. Each new letter represents a new feature. In spite of all its features, SARIMAX still can’t do transfer learning. This evolution has had less time to work on newer models like LSTM. LSTM wasn’t really designed for this type of task, but with a few modifications it could be. Improving its ability to deal with stationary time series, adding seasonality, and using moving averages could make a model that is better suited for this specific task.

Since each analysis has a limited time frame, it is impossible to try everything. There were many things that were considered and left untried due to this temporal reality. Some of the previously mentioned improvements are easy to try, but time consuming. Others require a lot of thought, time, and work to implement. When considering which ones to implement it is important to consider what was achieved in this analysis. SARIMAX was adequate for the task, but it is so complex that it requires more processing time than any other model. Maybe the focus should be on reducing the processing time of something that already works well.

# References

Brownlee, J. (2021, February 15). *Regression metrics for machine learning*. MachineLearningMastery.com. https://machinelearningmastery.com/regression-metrics-for-machine-learning/

Chauhan, N. S. (2021, January 2). Stock market forecasting/arima. Kaggle. https://www.kaggle.com/code/nageshsingh/stock-market-forecasting-arima

Frankel, M. (2023, December 6). *What is a P/E ratio?*. The Motley Fool. https://www.fool.com/terms/p/pe-ratio/

IBM. (2021, August 17). CRISP-DM help overview. IBM.

<https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>

IBM. (2024, February 23). *Rolling and moving averages*. <https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=dimensionally-rolling-moving-averages>

Kamenova, E. (2024, May 30). *Data Analyst Job Market in 2024: Demands and prospects*. 365 Data Science. <https://365datascience.com/career-advice/the-data-analyst-job-market/>

Ross, S. (2024, February 6). *How capital investment influences economic growth*. Investopedia. https://www.investopedia.com/ask/answers/032615/how-does-total-capital-investment-influence-economic-growth.asp

Mooney, P. (2023, January 16). *Stock market data (NASDAQ, NYSE, S&P500)*. Kaggle. https://www.kaggle.com/datasets/paultimothymooney/stock-market-data

OpenAI. (2024). ChatGPT (July 2 version) [Large language model]. https://chat.openai.com/chat

Pandas. (2024). Ecosystem. https://pandas.pydata.org/community/ecosystem.html

Paramartha, S. (2022, February 6). *Microsoft stocks price prediction*. Kaggle. https://www.kaggle.com/code/paramarthasengupta/microsoft-stocks-price-prediction

Resende, D. (2024, January 26). *Arima, Sarima, and SARIMAX explained*. Zero To Mastery. https://zerotomastery.io/blog/arima-sarima-sarimax-explained/

Sahoo, D. (2023, May 26). *Stock market index data India (1990 - 2022)*. Kaggle. https://www.kaggle.com/datasets/debashis74017/stock-market-index-data-india-1990-2022?select=NIFTY%2B100\_Data.csv

Sayah, F. (2023, January 31). 📊stock market analysis 📈 + prediction using LSTM. Kaggle. https://www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm

Scikit. *Sgdregressor*. scikit. (2025). https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.SGDRegressor.html

Shpagin, O. (2024a, January 2). *Huge US 514 stocks + 1298 columns market data 25GB*. Kaggle. https://www.kaggle.com/datasets/olegshpagin/extra-us-stocks-market-data

Stefano, A. D. (2023, June 15). *Machine learning in Stock prediction: A tech overview*. Machine Learning in Stock Prediction: a Tech Overview. https://www.itransition.com/machine-learning/stock-prediction

Wikimedia Foundation. (2024, July 20). *Long short-term memory*. Wikipedia. https://en.wikipedia.org/wiki/Long\_short-term\_memory