***Multiple Linear Regression Analysis on S&P 500 dataset***

***Western Governors University***

***By: Christian LeBlanc***

***Table of Contents***

***Research Question 3***

***Data Collection 4***

***Data Extraction and Preparation 6***

***Analysis 13***

***Data Summary and Implications 24***

***References 25***

***Research Question***

**A.**

The purpose of this analysis is to answer the research question: “Can a Multiple Linear Regression (MLR) model indicate which stocks in the “Big Tech” group have the strongest influence on the S&P 500 index?” This study will use MLR to analyze predictor variables.

Chen (2020) expresses how important and challenging statistical models are in analyzing the stock market. Warren Buffett is one of the most successful investors and he has stated the S&P 500 index is the best investment that most people can make (Frankel, 2024). This brings up an idea of can a subsection of the S&P 500 produce a better investment result. Technology has become such an engrained part of everyday life with the leaders in it becoming corporate giants. The multiple linear regression will show which of the 14 “Big Tech” stocks have the strongest effects on the S&P 500.

This study will contribute to the field of Data Analytics and the MSDA program by advancing the application of multiple linear regression techniques in financial markets. “Multiple linear regression is a method we can use to quantify the relationship between two or more predictor variables and a response variable” (Bobbitt, 2020). In this analysis the MLR model will quantify the fourteen stock predictor variables relationship with the S&P 500 index, the response variable. Ideally it will show a group of stocks that have strong positive coefficients to the S&P 500 index. Getting a group of stocks that have beaten such a good index in a business segment that will only continue to be a part of day-to-day life could result in having a good investment opportunity.

The Null hypothesis for the research question is “A MLR model does not show which stocks in the “Big Tech” group has the strongest influence on the S&P 500 index.” The alternative hypothesis is “A MLR model does not show which stocks in the “Big Tech” group has the strongest influence on the S&P 500 index.” This project aims to create multiple linear regression model that show the stocks in the “Big Tech” group that has the strongest influence on the S&P 500 index. “As of June 23, 2023, the average return of the top 7 stocks is 87%, while the average return from the rest of the S&P 500 is just 3%,” (Hammers, 2023). This quote supports the alternative hypothesis by showing that certain stocks have a greater influence on the S&P 500. A rejection of the null hypothesis would allow for more insight into the influence each individual stock in the group has.

***Data Collection***

**B.**

The data needed to be collected is for the S&P 500 index and stocks that make up that index and is available through the Kaggle website (Mvd, n.d). The dataset is updated on a daily basis on days the stock market is open. The data was given a usability rating of 10 out of 10 by Kaggle with 100% completeness, credibility, and compatibility. The original data contains both Quantitative and Qualitative variables. Overall data sparsity is < 5%.

<https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks>

The data is found in two different data sets. The raw data set on the stocks found in the S&P 500 has 8 columns and 1,832,429 rows and the data set on the S&P 500 index has 2 columns and 2,535 rows. The 8 columns in the raw stock data are: Date, Symbol, Adj Close, Close, High, Low, Open, and Volume. The 2 columns in the raw index data are Date and S&P500. The data requires some cleaning and transformation to get it ready for the model. The S&P500 is the dependent variable. The two data sets are publicly available for use on the website Kaggle.

|  |  |
| --- | --- |
| **Field** | **Type** |
| Date | Continuous |
| AAPL | Continuous |
| ADBE | Continuous |
| AMZN | Continuous |
| CRM | Continuous |
| CSCO | Continuous |
| GOOGL | Continuous |
| IBM | Continuous |
| INTC | Continuous |
| META | Continuous |
| MSFT | Continuous |
| NFLX | Continuous |
| NVDA | Continuous |
| ORCL | Continuous |
| TSLA | Continuous |
| S&P500 | Continuous |

One advantage of using this data is it is complete data. The data has information for all the columns for each date that a stock is in the S&P 500. This raw data contains all the information needed for the analysis without the need to include imputations. Avoiding the need to input missing values helps to keep bias or inaccuracies out of the model. It ensures that the model is built on the most accurate and representative dataset possible. This enhances the overall quality and reliability of the regression model.

One disadvantage the data-collection process had was the format of the data. The data had to be manipulated to get it in the model. The dataset had all the information needed, but it is possible that another dataset would have all the information in a format that would have been easier to prepare it.

***Data Extraction and Preparation***

**C.**

The data extraction and preparation begins with importing the packages and libraries that were used to complete a previous MLR. Using pandas, both csv files are imported into the Jupyter notebook.

A screenshot of a computer program

Description automatically generated

Using .info() to confirm both of the data sets are as expected.

A screenshot of a computer

Description automatically generated

Filtering out the rows to only keep the lines that are pertaining to the stocks that are a part of this analysis. Using .info() to examine what the filtering did to the data. This shows the presence of null values.

A screenshot of a computer program

Description automatically generated

Pulling up a sample of the rows with null values. The META stock is the first one that is shown with the null values, so it will get investigated first.

A screenshot of a computer

Description automatically generated

Meta was not a publicly traded stock during the dates that were shown. Dropping all the dates before May 18th, 2012. To do this the column “Date” had to be changed from object data type to datetime64. In dropping all rows with dates before this does two things. It should remove all the nulls associated with the Meta stock and allow all the stocks to have the same number of dates when using them in the model. After doing this, .info() is used to check the non-null counts and all the columns have the same amount of non-null as the number of entries so this took care of the nulls in the data.

A screenshot of a computer

Description automatically generated

Checking to see the number of unique values in the “Symbol” column are fourteen as this analysis is focused on them.

A computer screen shot of a computer code

Description automatically generated

Changing the date in the data set for the S&P 500 index to be able to drop the same dates as the data set with the stocks. After doing this, .info() is checked again. It is noticed that nothing was dropped. The data in the index data set shows that it starts on May 30th, 2014.

A screenshot of a computer code

Description automatically generated

After the start date on the index was noticed, the stock data set needs to be updated to match the index data set.

A screenshot of a computer

Description automatically generated

Now that the data has the same date range, pandas is used to merge the two. Checking .info() to make sure all the expected columns are present and do not have nulls present.

A screenshot of a computer

Description automatically generated

The data is now all together, but it needs to be reshaped to get it in the model as needed. Using the pivot feature a new data set was able to be formed. This new data set has an index column of the dates, the columns are the fourteen unique stocks from “Symbol”, and “Adj Close” is used to fill in the values. The column for the values of the S&P 500 index is added back to show its value on the dates.

A screenshot of a computer screen

Description automatically generated

The .info() command is used to check one more time that the new data set is in the format needed to run the initial model.

A screenshot of a computer

Description automatically generated

Python in a Jupyter notebook was used to do the data extraction and preparation. The packages and libraries in Python that were imported are pandas, numpy, matplotlib, seaborn, scipy, and statsmodels.api. The pandas library allowed for methods pd.read\_csv, .info(), .isin(), .isnull(), pd.to\_datetime, pd.Timestamp, .nunique(), pd.merge, .pivot\_table, and .head() to be used.

The method pd.read\_csv is used to grab the csv files with the data sets needed for the analysis. Using .info() shows the total entries, columns, non-null count, and the data type. The method .isin() is used to assign just the lines with the stock symbols that are needed in this analysis. To investigate the nulls that were present the method .isnull() is used. The column Date needed to be used to filter by the date. Using pd.to\_datetime is used to change it from object data type, then pd.Timestamp to state a date to cut the data from before that date. The number of unique values of stocks needed to be fourteen. The method .nunique() is used to check this is the case. The two data sets needed to be combined to do the MLR, so pd.merge was used to do this merging on the column “Date”. The method .pivot\_table is used to form a table with the fourteen stocks, S&P500, and Date as the columns and the values coming from the adj closing price. The method .head() is used to double check the pivot table is formatted in a way that would work in the MLR.

Jupyter Notebook provides an interactive environment ideal for exploratory analysis. It allows for immediate feedback and visualization, which is crucial for data preparation and understanding the dataset. The combination of the tools and techniques used offers a robust framework for data extraction, preparation, and analysis. They provide a balance of efficiency, functionality, and ease of use, essential for handling complex datasets.

An advantage to using Python in a Jupyter notebook to do the data extraction and preparation is built in reference materials from having used the similar processes in previous regression models.

A disadvantage to using the data extraction and preparation techniques used above is having to repeat steps. Filtering out the dates multiple times is an example of this disadvantage causing the analysis to take longer.

***Analysis***

**D.**

Matplotlib.pyplot library is used to make the univariate and bivariate visualization. The univariate visualization shows the prices of the stock or index over time in a line graph. Joint plots are used to compare the price of the S&P 500 index price to the price of each of the fourteen stock prices. The MLR to be used is the Ordinary Least Squares (OLS). This requires the library of statsmodels.api to be imported. Once the initial model is established, Variance Inflation Factor (VIF) will be used to eliminate the predictor variable with highest VIF that is over the acceptable value of 10 for this analysis. This will be reproduced until all predictor variables are in the acceptable range, in hopes of this handling any multicollinearity. Backward stepwise elimination will be used to eliminate any features that do not meet the criteria for being statistically significant. The significant level for this analysis is 5%, which will be P-values under 0.05 are acceptable. The model will then be checked for the assumption of Normality of Residuals. A Q-Q plot will be used to visually inspect this.

The first calculations are line graphs of each of the fourteen stock prices and the S&P 500 index price over time. The univariate visualizations are similar to looking at each’s stock price history, but it is narrowed to focus just on the dates that will be in the model.

A collage of graphs

Description automatically generated

Next the bivariate visualizations of the fourteen stocks to the S&P 500 index. Joint plots are used to the relationship of the prices and the distribution of each.

A graph with blue lines

Description automatically generated with medium confidence

A graph with blue squares

Description automatically generated with medium confidence

A graph of different sizes and shapes

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

A graph of different sizes and shapes

Description automatically generated with medium confidence

The initial Multiple Linear Regression model is created now. The model is created with the dependent variable being “S&P500”. The independent variables are the fourteen stocks and a constant. The model used to create this initial model is OLS.

A computer code with red and white text

Description automatically generated

The results of the initial model show that the presence of multicollinearity may be an issue with the model.

A screenshot of a computer

Description automatically generated

Getting the Residual Standard Error (RSE) of the initial model. This result is high. The RSE will be calculated again after the model is ran with multicollinearity is reduced through VIF and each feature is confirmed to be statistically significant.

A close-up of a message

Description automatically generated

The Variance Inflation Factor (VIF) is found on the variables used in the initial model. The highest value of 440.35 for the variable “MSFT” is well over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 281.86 for the variable “ORCL” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer program

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 220.86 for the variable “GOOGL” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 186.50 for the variable “CRM” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 138.47 for the variable “ADBE” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer program

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 119.48 for the variable “AMZN” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer code

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 82.63 for the variable “CSCO” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer code

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 70.57 for the variable “META” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer program

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 53.63 for the variable “INTC” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer code

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. The highest value of 52.14 for the variable “AAPL” is over the limit of 10 used for this analysis. This variable will be removed and the VIF will be ran again.

A screenshot of a computer

Description automatically generated

The VIF is now ran with the previous highest variable over the threshold removed. All variables are now under the 10.

A screenshot of a computer program

Description automatically generated

Now that the VIF is showing the remaining variables with acceptable values, another model will be run. This model is meant to be the beginning of using Backward Stepwise Elimination to ensure all features are statistically significant. The p-values are under the 0.05 value threshold for all variables, so this is not needed.

A screenshot of a computer

Description automatically generated

Getting the RSE for the new model.

A close-up of a message

Description automatically generated

A Q-Q plot is ran to get a visual on the assumption of normality.

A graph with a red line

Description automatically generated

A heatmap is ran to visually check the correlations between the remaining variables that are in the model.

A screenshot of a computer screen

Description automatically generated

Python was used for the creation of the regression model. According to Data Scientist’s Analysis Toolbox: Comparison of Python, R, and SAS Performance (Brittain et al., 2020) Python was the faster than both R and SAS when processing a regression model, doing so in 1.15 seconds. “We see the market slightly bending towards Python in today’s scenario,” (Jain, 2020). The article is comparing Python, R, and SAS in the world of data science and it shows that it is trending towards Python being more useful of a tool.

An advantage to using Python to create the MLR model is speed at which it processes a regression model, as pointed out earlier. The libraries are significant in the speeds. Statsmodels is optimized for speed and efficiency in doing statistical computations.

The assumptions that are made with using OLS is a disadvantage to this technique. The model relies on several assumptions to ensure that the results are reliable. Violations of the assumptions can result that are inconsistent and inefficient.

***Data Summary and Implications***

**E.**

The analysis failed to reject the null hypothesis. A MLR model was not able to show which stocks in the “Big Tech” group has the strongest influence on the S&P 500 index. The model still has too many issues to be used reliably. Using the VIF to drop features until only the 4 of the 14 remained did not improve the model. Despite using VIF to reduce multicollinearity, the model with 4 features still produced a high condition number. The original model had a lower Akaike Information Criterion (AIC). It went from 2.905e+04 to 3.372e+04. The RSE grew from 74.34 to 186.98 showing the initial model had a better fit. The Q-Q plot indicates that the residuals do not follow a normal distribution, as the points deviate significantly from the reference line. This deviation suggests potential issues with the assumptions of normality.

One limitation on the analysis is using a small selection of the 500 stocks that make up the S&P 500 index. The analysis was focused on a very small group of stocks and interested in the impact they have, but could useful information come out of a MLR that had all 500 in the initial model.

It is recommended to not use this model for any insights. This is still a topic that would be interesting to get useful information from. There are two different directions further analysis should investigate. One would be to use more features. It would be interesting if including all 500 would help the model to become useful. The other direction would be to find other metrics in the economy that are not stocks and see if a MLR can be built and useful.

**F.**

**Sources**:

Bhandari, A. (2020, March 18). What is multicollinearity? Analytics Vidhya. Retrieved from July 11, 2024, from

<https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/>

Bobbitt, Z. (2020, November 18) Multiple Linear Regression by Hand. Statology. Retrieved from July 24, 2024, from

<https://www.statology.org/multiple-linear-regression-by-hand/>

Brittain, J., Cendon, M., Nizzi, J., & Pleis, J. (2018). Data scientist’s analysis toolbox: Comparison of Python, R, and SAS

performance. SMU Data Science Review, 1(2), Article 7. Retrieved from July 11, 2024, from

<https://scholar.smu.edu/cgi/viewcontent.cgi?params=/context/datasciencereview/article/1021/&path_info=auto_convert.pdf>

Chen, S. (2020, May 12). Forecasting daily stock market return with multiple linear regression. Retrieved from July 11, 2024,

from <https://digitalcommons.latech.edu/cgi/viewcontent.cgi?article=1015&context=mathematics-senior-capstone-papers>

Frankel, F. (2024, March 26). What is the S&P 500 index? The Motley Fool. Retrieved from July 11, 2024, from

<https://www.fool.com/investing/stock-market/indexes/sp-500>

Hammers, J. (n.d.). The impact of the S&P tech industry concentration. TFG Invest. Retrieved from July 11, 2024, from

<https://www.tfginvest.com/insights/the-impact-of-the-sp-tech-industry-concentration>

Jain, K. (2017, September 12). SAS vs. R vs. Python: Which tool should I learn? Analytics Vidhya. Retrieved from July 11, 2024,

from <https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/>

Mvd, A. (n.d.). S&P 500 stocks [Dataset]. Kaggle. Retrieved from June 26, 2024, from

<https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks>

Sonsini, J. (2022). Adjusted close vs. close. Kaggle. Retrieved from July 11, 2024, from

<https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks/discussion/364874>