Seshagiri Rao Mallina A16659161, Neil Sharma A17579739, Kelly Chang A16406092
DSC 148
17 March 2024

Final Project Report: Predictive Model for Top Heaviness in the Economy

Introduction:

Understanding the influence of leading companies on the broader market is crucial, particularly within indices like the S&P 500 where the performance of a handful of high-cap stocks can significantly impact the overall movement of the market index. Our project focuses on examining this dynamic by leveraging daily stock trading data of both top-tier and lower-tier companies within the S&P 500.

The goal is to ascertain the extent to which the top companies sway the index and, by extension, affect the stocks of smaller companies within the same index. We will use different predictive models to see if we can predict how the performance of top companies affects the performance of the bottom companies. Our analysis will highlight the distribution of influence within the S&P 500 and also enable a deeper understanding of market structure. Through this project, we're setting out to provide a clearer picture of the 'top-heaviness' of the S&P 500 and its implications for the broader market.

1 Dataset:

1.1 Identify Dataset

The dataset compiled for this analysis was sourced from the Yahoo Finance API. Through API calls, we gathered data on specific segments of the S&P 500 index: the "Magnificent 7", which are the top seven performers, which include: Google (GOOG), NVIDIA (NVDA), Amazon (AMZN), Meta Platforms (META), Tesla (TSLA), Apple (AAPL), Microsoft (MSFT). We also gathered data on the SPY ETF representing the market performance of the S&P 500 as well as the bottom 20 stocks ('ETSY', 'FMC', 'FRT', 'GNRC', 'HAS', 'IVZ', 'MHK', 'MKTX', 'NCLH', 'PARA', 'PNW', 'RHI', 'VFC', 'WHR', 'XRAY', 'ZION') were identified based on. From 2015 to 2024, the dataset encapsulates nine years of stock data, offering a long timeframe for analysis. It comprises 15,862 observations for the "Mag 7", 2,286 for the SPY ETF, and 36,234 for the bottom 20 stocks, totaling over 50,000 observations. The dataset includes daily stock information, such as: opening price (Open), the highest (High) and lowest (Low) prices of the stock within the trading day, the closing price (Close), the adjusted closing price (Adj Close) accounting for stock splits and dividends, and the trading volume (Volume).

2 EDA:

2.1 Data Cleaning:

Top 7 S&P 500 companies were extracted within a set date range of 9 years of the most recent code run date. The stock data frames were then concatenated into a single data frame. The bottom 20 companies were processed in the same way. Two additional data frames were made for future exploratory analyses: top 7, bottom 20, and SPY concatenated, and the top 7 data frame concatenated with bottom 20 data frame. We reset the index into a default integer index, and renamed this newly created column as 'Date'. We added a 'Returns' column across the dataset, calculated as the percentage change in the adjusted closing price ('Adj Close') multiplied by 100. We augmented the dataset for each of the bottom 20 companies by integrating the 'Returns' and 'Volume' data from each of the 'Mag 7' companies and the SPY ETF. The final dataset was prepared without dropping any columns or rows. 'Returns' and 'Volume' were used because these two features were the most relevant in understanding the performance and sentiment of that day.

2.2 Trends/Findings:

We explored all of the columns in the dataset. Below are columns that we closely investigated and are worth discussing.

2.3 Volume:

Amongst the top 7 stocks and SPY, Tesla's stock volume had higher peaks than other stocks, with Apple following close behind. The bottom 20 stocks have a much milder fluctuation, with the most dynamic stocks. PARA and NCLH, peaking at around a quarter of Apple and Tesla's volume. 2020-2022 peaks indicate that the onset of the COVID-19 pandemic caused volatility in financial markets.

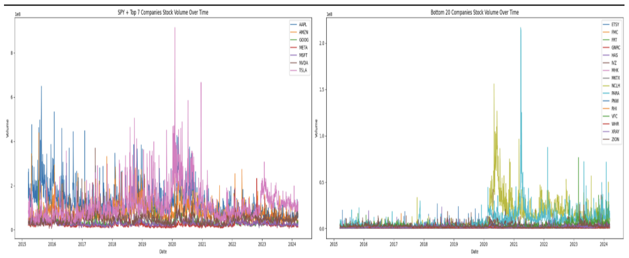


Figure 1: This figure visualizes the volume of each stock over time.

2.4 Adjusted Close:

The adjusted close indicates the value of a stock over time by modifying its historical prices to reflect timewise changes. While all stocks observed a general increase in valuation over the last 9 years, Nvidia experienced a sharp rise after 2023. GNRC and MKTX stocks experienced high valuation in 2021, then experienced sharp declines by 2023 possibly from external factors like COVID.

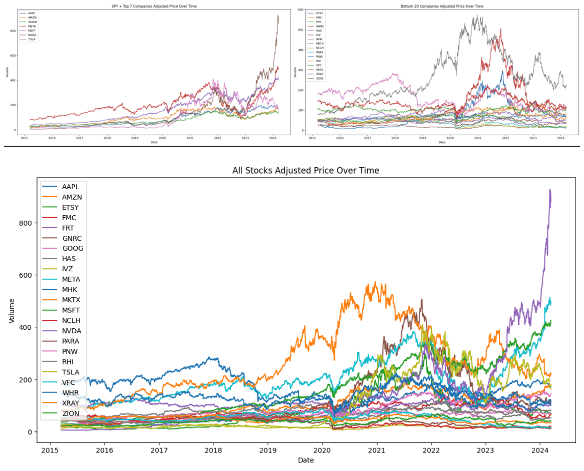


Figure 2: Adjusted prices of the top 7 stocks plus SPY (top left) and that of the bottom 20 stocks (top right).

2.5 30-Day Moving Average:

The 30 day moving average creates a constantly updated average price. We are able to more easily identify the underlying trends beyond daily volatility. Our conclusions are similar to our adjusted prices over time plots in Figure 2, where Nvidia shows an uptrend, suggesting that the stock has been steadily gaining momentum in the past year. MHK seemed to be losing momentum up until 2020, where the company experienced a slow increase in 2021 before dropping. The top 7 companies and SPY seemed to see a steady incline in valuation over the years, and did significantly better in the past 2 years, the bottom 20 companies see volatile trends. MKTX's peak is higher than the peaks of some top 7 company peaks, which may indicate that it has a strong position in the market, and may be less influenced by the top 7 stocks.

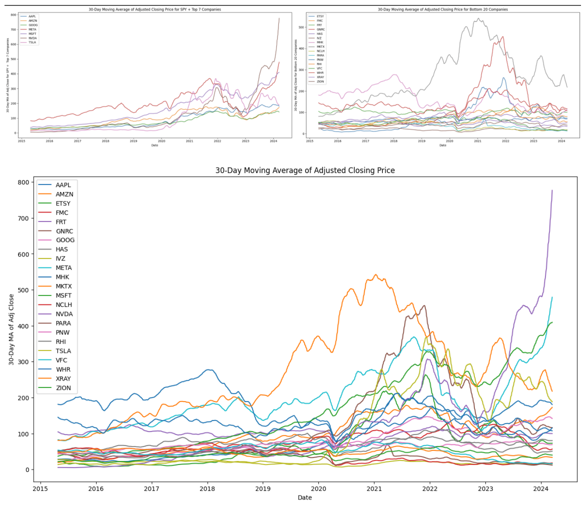
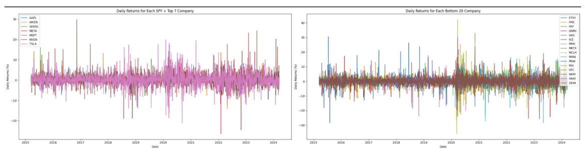


Figure 3: 30 day moving average of adjusted closing prices over time shows similar trends as Figure 2.

2.6 Returns

Returns measure the percentage change in the stock price from one trading day to the next, giving us insights into the short-term performance and volatility of a stock. NCLH's stock had the most fluctuation in return, with a maximum change in stock price of 80. Looking at the overall trends, we see a higher concentration of volatile stocks after the beginning of 2020, which could potentially be attributed to the effects of COVID-19.



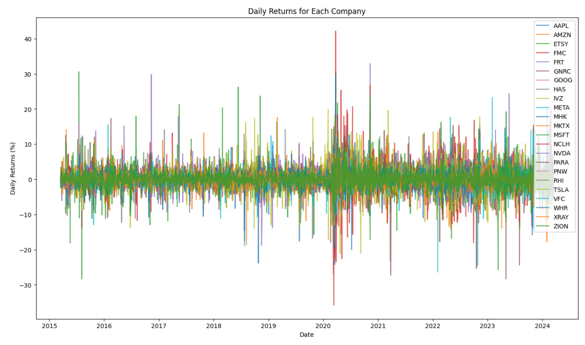


Figure 4: Daily returns for each company are shown in the three plots above.

2.7 Custom Splicing Function:

The splice_data function divides the dataset into training and testing sets, tailored for our predictive modeling task. Our function is called with parameters specifying the start and end dates of the period, along with the desired lengths of time for training and testing. For example, to utilize data from January 1, 2023, to March 16, 2023, we would set the start date to '2023-01-01' and the end date to '2023-03-16', covering a total of 75 days. This period is segmented into the first 60 days for training and the last 15 days for testing. To construct our train and test datasets, we segmented the data frame into two portions: one for training and the other for testing. For the training set, we extracted columns corresponding to the 'Mag 7' volume and returns, as well as those for the SPY volume and returns, thus forming two separate training datasets. This process was mirrored for the test data. The returns from the bottom 20 companies were designated as the dependent variable, shaping our 'test_set' and 'y_train'. Consequently, this systematic approach yielded two training sets ('train_mag_7' and 'train_spy'), two testing sets ('test_mag_7' and 'test_spy'), and the dependent train and test sets ('y_train' and 'test_set').

3 Predictive Task:

Our predictive task is to use the trading volume and returns of the "Mag 7" S&P 500 companies as one set of predictors, and the trading volume and returns of the S&P 500 ETF (SPY) as another, to predict the returns of the bottom 20 companies within the S&P 500.

3.1 Evaluation Metric:

We selected Root Mean Square Error (RMSE) as our evaluation metric. RMSE quantifies the difference between values predicted by a model and the values observed. Its utilization is particularly apt for our project, given its ability to impose more significant penalties on larger errors than on smaller ones. Given that our returns data are of relatively small magnitude, and predictions are made over a sizable dataset, a smaller RMSE value will be indicative of better model performance.

3.2 Baseline:

For our project, we established a baseline using four models (to predict the returns of the bottom 20 companies in the S&P 500, using both the "Mag 7" and SPY ETF as predictors.

We chose the Linear Regression model since it operates under the assumption that there's a linear association between our independent variables and our dependent variable. Its inherent simplicity and reduced risk of model overfitting provides a clear benchmark.

Random Forest Regressor is capable of capturing non-linear relationships between features and the target variable, working well with both categorical and continuous data.

XGBoost assembles weak learners and aims to achieve high accuracy, offering an advantage in forecasting financial returns.

SARIMAX incorporates predictors into the time series forecasting process, providing a more detailed analysis of how the "Mag 7" and SPY influence the returns of the bottom 20 companies.

All models were trained on the same set of features extracted from the processed data and evaluated on identical test datasets. The selection of intervals for training and testing was made to ensure a comprehensive coverage across all years in the dataset. To maintain the integrity of our comparison and avoid potential overfitting, we opted for default hyperparameters for each model.

3.3 Baseline Result:

Linear Regression:

	Interval	RMSE Mag_7	RMSE_SPY
0	2023-10-01 (+75d train, +15d test)	28.594580	23.779139
1	2015-06-01 (+180d train, +30d test)	25.674886	23.153984
2	2018-03-01 (+120d train, +14d test)	18.154962	16.325983
3	2016-09-01 (+90d train, +30d test)	27.203700	20.775566
4	2019-11-01 (+60d train, +30d test)	27.842797	21.693390
5	2020-06-01 (+200d train, +30d test)	38.482021	34.562233
6	2021-04-01 (+60d train, +15d test)	33.455954	26.607050
7	2022-07-01 (+80d train, +10d test)	39.214553	32.489637
8	2019-02-01 (+365d train, +100d test)	77.850166	67.081261
9	2017-02-01 (+30d train, +5d test)	25.116692	14.150226

Table 1: Predictive performance results for Linear Regression Model.

Random Forest:

	Interval	RMSE Mag_7	RMSE_SPY
0	2023-10-01 (+75d train, +15d test)	29.323161	29.853578
1	2015-06-01 (+180d train, +30d test)	25.737018	26.467290
2	2018-03-01 (+120d train, +14d test)	22.170341	19.605293
3	2016-09-01 (+90d train, +30d test)	21.828434	24.413586
4	2019-11-01 (+60d train, +30d test)	23.121113	24.540656
5	2020-06-01 (+200d train, +30d test)	37.421914	38.806210
6	2021-04-01 (+60d train, +15d test)	28.125734	27.894164
7	2022-07-01 (+80d train, +10d test)	35.965360	37.570322
8	2019-02-01 (+365d train, +100d test)	86.680326	79.987210
9	2017-02-01 (+30d train, +5d test)	18.416151	17.331637

Table 2: Predictive performance results for Random Forest model.

XGBoost:

	Interval	RMSE Mag_7	RMSE_SPY
0	2023-10-01 (+75d train, +15d test)	32.403935	40.679375
1	2015-06-01 (+180d train, +30d test)	27.921141	29.918283
2	2018-03-01 (+120d train, +14d test)	32.284965	21.730372
3	2016-09-01 (+90d train, +30d test)	24.435928	26.645434
4	2019-11-01 (+60d train, +30d test)	25.119079	27.518631
5	2020-06-01 (+200d train, +30d test)	43.025072	45.060319
6	2021-04-01 (+60d train, +15d test)	31.257761	31.286528
7	2022-07-01 (+80d train, +10d test)	41.638075	41.138969
8	2019-02-01 (+365d train, +100d test)	91.064021	82.272886
9	2017-02-01 (+30d train, +5d test)	21.462523	21.787445

Table 3: Predictive performance results for XGBoost.

SARIMAX:

	Interval	RMSE Mag_7	RMSE_SPY
0	2023-10-01 (+75d train, +15d test)	128.8232287387416	72.92168903593674
1	2015-06-01 (+180d train, +30d test)	42.63544296823535	38.58234821092777
2	2018-03-01 (+120d train, +14d test)	43.376081983053766	26.74314362042826
3	2016-09-01 (+90d train, +30d test)	85.12223451888298	42.90838472877898
4	2019-11-01 (+60d train, +30d test)	82.45890380736601	36.28935523805701
5	2020-06-01 (+200d train, +30d test)	69.05960007229892	48.566539062389054
6	2021-04-01 (+60d train, +15d test)	61.92340233891537	36.76792204113141
7	2022-07-01 (+80d train, +10d test)	65.6768847409059	53.134069010030395
8	2019-02-01 (+365d train, +100d test)	93.23901959865452	73.83605895702922
9	2017-02-01 (+30d train, +5d test)	83.85363391993195	57.37979417203162

Table 4: Predictive performance results for SARIMAX.

3.4 Baseline Conclusion

It's evident that neither indicator outperforms the other across all models and time intervals. However, there's a trend where the "Mag 7" indicator tends to yield lower RMSE values during stable market periods. The tables show that the "Mag 7" indicator typically yields lower RMSE values in more stable economic periods, such as the interval starting from 2019-11-01, where we observe the RMSE for "Mag 7" being consistently lower than that for the SPY across all models. This trend suggests that during times without market disturbances, the performance of these top companies is a more accurate predictor of the bottom 20 companies' returns. In contrast, during the volatile period of early 2020, which encompasses the onset of the pandemic, the SPY indicator tends to perform better than "Mag 7". In the interval starting from 2020-06-01, the RMSE for SPY is relatively lower or close to "Mag 7" reflecting the broader market's influence under turbulent economic conditions. This pattern indicates that during times of widespread market shifts, an ETF can serve as a more reliable indicator for forecasting. In conclusion, the choice between "Mag 7" and SPY as a more reliable predictor may depend on the specific temporal context, with "Mag 7" being more suitable for stable periods and SPY during times of market upheaval. “Mag 7” also performed better during shorter intervals, achieving lower RMSE than SPY when intervals were shorter than 100 days, indicating “Mag 7’s” role as a short term indicator and SPY’s role as a better long term indicator.

4 Deep Learning Model:

A different approach from the baseline models was used for the deep learning model. Predicting on numerous intervals would create inconsistent input shapes, making it harder for the model train in theory. To work around this, we created splices of three day intervals on two different datasets. The first dataset had the daily stats for each of the bottom 20 stocks and the corresponding SPY returns and volume for that day. The second dataset had the Top 7’s returns and volume instead of SPY. The first two days would be our X, and the third day’s returns would be our target.

The X was a tensor in the form:

[*batch_size,* 1*, num_days*=2*, num_features*]

The y data, which was the returns of the stock the next day was a tensor in the form:

[*batch_size,* 1]

As shown, our X is multi-dimensional based on how many days we choose to feed into the model.

Since the data is multi-dimensional, we opted for an unconventional approach to our network architecture, utilizing a convolutional neural network. Even though our data is not image based, we theorized that since CNNs are adapted to work with multi-dimensional data, we wouldn’t have to lose multiple days of context at the expense of model performance. The ideal CNN would've been complex, like ResNet128, but due to computational constraints, we created a simple CNN architecture to train our models. Hyper parameters such as learning rate and batch size were assigned manually.

4.1 Model Optimization:

While MSE is not an ideal loss choice for this problem, in order to compare the CNN model to the baselines that use RMSE loss, MSE was maintained for the training of our two models. Along with our criterion, we chose Adam as our optimizer, since it pairs well with CNNs.

Due to the complexity of the problem, we implemented a learning rate scheduler based on the validation loss during training. A scheduler is necessary for a problem this complex as it helps the model avoid local minimas and make training more stable, allowing the model to converge faster. The learning rate was reduced by a factor of 10 every 10 epochs or when a new minimum validation loss was reached after epoch 25. We trained both the SPY and “Mag 7” model on 100 epochs, where SPY had 25 features and “Mag 7” model was trained on 37 features.

4.2 Model Results & Takeaways:

Our two main models achieved significantly lower loss values than our baselines. Our SPY CNN model achieved a minimum validation MSE loss of 4.206 and an accuracy of 51.4%. Our “Mag 7” CNN model achieved a minimum validation MSE loss of 4.209 with an accuracy of 50.7%. Accuracy was measured by whether the sign of the prediction and target were the same. The CNN models heavily outperformed our baseline models, achieving a higher accuracy, lower validation loss, and more realistic outputs.

Our attempts, while providing us with valuable insights and decent performance, could be improved upon. Our architecture was too simple in order to effectively capture patterns within the data, commonly converging at around epoch 30. A custom loss function (further explained in Limitations) would have to be paired with said architecture to help the model understand the importance of accuracy in sign. If given more time and computational resources, implementing these corrections would greatly improve performance.

5 Results & Conclusion:

Throughout our baselines and main models, SPY was shown to be a slightly better predictor for the bottom 20 stocks than the "Mag 7". However when the intervals were shorter, "Mag 7" was equal to, if not a better indicator, than SPY. In the CNN model, the intervals were constant and extremely small, and both models achieved very similar results. In the baseline models, baseline results where the interval was less than 100 days had the "Mag 7" be a better indicator for stock performance than SPY by a significant margin.

With our results, we can conclude that while SPY as a whole is more often than not a better indicator for companies within the ETF, the performance of the top 7 can serve an important role when analyzing short term performance. While none of the models were good enough to predict accurate returns, the outputs of the models can provide a good baseline for how stocks should perform based on recent data from SPY and the "Mag 7".

6 Limitations:

There are a multitude of limitations this project faces, particularly when it comes to context not explicitly given by the dataset. Identifying if the current data is during a bull or bear market, if the date serves a particular purpose such as an earnings report release, a federal interest rate change, or a political event, etc. These limitations, however, would require extensive labor to take into account for, and are not practical given the context of this project.

We identified a few, more practical, areas of our project that could warrant further exploration. First, we did not include the Market Group Adjuster (MGA7) effect, which adjusts the market cap of companies to reflect their free-float market capitalization. Future research would incorporate this adjuster to enhance the accuracy of our assessment of company influence within the index. The market cap adjustments could have potential impact on the relationships between top and bottom companies in the index, as well as the overall market movement forecasts. Additionally, our project encountered limitations due to computational constraints, which impacted our ability to employ more complex predictive models. Having resources with more computing power would improve our understanding about the influence of top companies on the rest of the S&P 500, and generate more accurate predictions. For example, we had planned to use ResNet architectures and transformers for analyzing temporal financial data, but they were too computationally expensive for us. Given more computational resources, future studies could explore beyond machine learning techniques, such as natural language processing for sentiment analysis, reinforcement learning for simulating market scenarios, and custom deep learning models for financial market predictions. Along with an increase in computational resources, we theorize that a custom loss function that penalizes the model harshly for having the wrong sign (predicts the stock goes up when it actually goes down, or vice versa) while still measuring the distance between the target and prediction would perform better. However, due to time constraints, this was not able to be implemented properly. Future studies could perform comparative analysis across different markets and regions, providing a global perspective on market cap influence. Examining similar dynamics in other indices and markets could help understand if the patterns we observe in the S&P 500 are unique to its own, or apply to a broader market as well.

7 Literature:

7.1 Baseline Models Inspiration

https://www.sciencedirect.com/science/article/pii/S1877050920304865?ref=cra_js_challenge&fr=RR-1
This paper inspired our project's baseline and custom models. It discusses previous works on machine learning in quantitative finance, particularly the usage of non-linear models for forecasting future asset values over linear models. We drew inspiration from the paper by varying length of time intervals in our baseline models, and added additional analysis based on seasons and pandemic circumstances. Like the methodology from the paper, we used SARIMAMAX as one of our baseline models (the paper uses ARIMA; SARIMAX is its counterpart for seasonal data), and found that our custom neural network performed significantly better than SARIMAMAX.

We were unable to compare our preprocessing as this was not shown in the paper, however we have the same data source and presumably handled data in similar ways. In our project, we experimented with both linear regression and non-linear models XGBoost/Random Forest. Contrary to what we expected, SARIMAMAX consistently generated larger RMSE compared to all three baseline models, of which all had approximately similar RMSE.

7.2 Forecasting Models
https://www.kaggle.com/code/avikumart/timeseries-stock-price-analysis-forecasting
Time Series Stock Price Analysis Forecasting by Avikumar T.
Avikumar's project intends to forecast stock prices with time series data. He has significantly less missing data, and thus did not require unique splicing functions to account for missing time values. Like our project, this project aims to do a comparative analysis with 7 tech stocks. He generated daily returns and mean growth, as well as a candlestick chart on the open-high-low-close chart (OHLC) for each stock. After his initial exploratory data analysis, he found moving averages of stocks for 10, 20, and 50 day windows, and forecasted using Prophet. Avikumar uses Auto-ARIMA as his main model, and found a slow decline in autocorrelation, indicating that the time series is stationary. Partial autocorrelation results show that Apple and Facebook stocks, and likely he infers that other top 7 tech stocks, have time series that can be captured with 1 lag, meaning the current value of a stock is correlated with its immediate previous day. The RMSE were both below 10 for both Facebook and Apple, indicating good forecasting. Similar to the results of our EDA, short-term forecasts elicited a more accurate prediction than longer-term forecasts. From this project, we learned about the usage of moving averages and the possibility of forecasting using Prophet for data with strong seasonal trends.

8 Code:
The code can be found in this github:
https://github.com/smallinaUCSD/DSC148FinalProject
yahoo.ipynb contains our data preprocessing.
CNN.ipynb contains our model code.

9 References:
[1] Time Series Stock Price Analysis Forecasting by Avikumar T.
https://www.kaggle.com/code/avikumart/timeseries-stock-price-analysis-forecasting
[2] Stock Market Prediction Using LSTM Recurrent Neural Network by Adil Moghar, Mhamed Hamiche
https://www.sciencedirect.com/science/article/pii/S1877050920304865?ref=cra_js_challenge&fr=RR-1