Predicting the S&P 500: A Not So Random Walk Down Wall Street

Anya Lauria, Alice Lu, Andrei Sebald, Erik Jarvis

University of California, San Diego

Abstract

This report explores and combines two different algorithms for predicting the S&P500, a stock market index that displays the performance of 500 different large companies. The first method utilized is ARIMA (autoregressive integrated moving average), an algorithm that takes note of the trend of one feature and continues that trend into the future to make predictions. The second method is a Random Forest which utilizes 3 select features to make predictions on given dates. We found that three other features most heavily impact the stock index: the US/China Foreign Exchange Rate, Europe/China Foreign Exchange Rate and US Treasury Bond Data that is used to calculate interest rates. By both leveraging these models individually, we will be able to accurately predict the value of the S&P 500 for any point in the test data set.

Introduction

The ability to predict the stock market is one sought after by every teenager with a Robinhood app, every stock broker on wall street, and all people in between. Books and programs will offer promises of obvious tells within the trends that you can use to predict rises or drops and quickly make millions. But as the old saying goes "if it were that easy, everyone would be doing it." There must then be some unpredictability to the stock market. Maybe it's as random as flipping a coin every minute to decide whether the price goes up or down. Maybe it follows a trend. Maybe there are other variables we can look at to predict what's going to happen to a certain stock. These are the uncertainties that cloud the stock market crystal ball. Even though the world is more or less stumped by this problem, we, a group of undergraduates at UCSD are pretty sure we know the answer– and we may have figured it out in just 36 hours.

Methodology

Dataset, Data Cleaning and Transformation

- **observations_train:** This dataset contains data on 68 unique series **including SP500** up until 2019. For each row in the dataset, there is a series id, the value of the series at the date it was collected and the date the value was collected. We filled in the missing values with the average of the surrounding values. Since some series were collected on a weekly and monthly basis, we filled in the missing values between each month/week with a **time interpolation** method, meaning that the missing values were filled by predicting the value at the time based on the assumption that the values increase/decrease linearly. We

also **pivoted** the data such that each series was arranged in columns in the dataframe. Finally, we **scaled** the data being used for the Random Forest regressor as each feature's value was in different units.

- **observations_test:** This dataset contains data on 67 unique series **excluding SP500** but including 2 new series not in the observations_train data set continuing from 2019. For each row in the dataset, there is a series id, the value of the series at the date it was collected and the date the value was collected. We filled in the missing values using the same methods used on observations_train.
- **series.csv:** Describes the different series, frequency, units and whether they are seasonally adjusted.

Feature Selection

Out of the 67 available features (excluding SP500) from both the training csv and test csv, we chose to **analyze 20 features** since only these 20 series had dates that matched up with SP500. Using a Random Forest Regressor, we determined that there are three best features:

- **BAA10Y (Treasury Bond Data from Treasury Dept.)**
- **DEXCHUS (China/Us Foreign Exchange Rate)**
- **DEXUSEU(Europe/Us Foreign Exchange Rate)**.

To prevent overfitting, our Random Forest Regressor model uses only these three features. There are also **high Pearson correlations** between these three features and S&P500 as seen in figure

1.1. Since all of these factors heavily impact every major business in the United States, it is understandable why these features are the most highly correlated with S&P500's stock index. The intersection between SP500 and our best features are colored the darkest, meaning there is high correlation. As for the ARIMA model, the only feature that is relevant is the data for S&P500 itself, due to the nature of the algorithm.

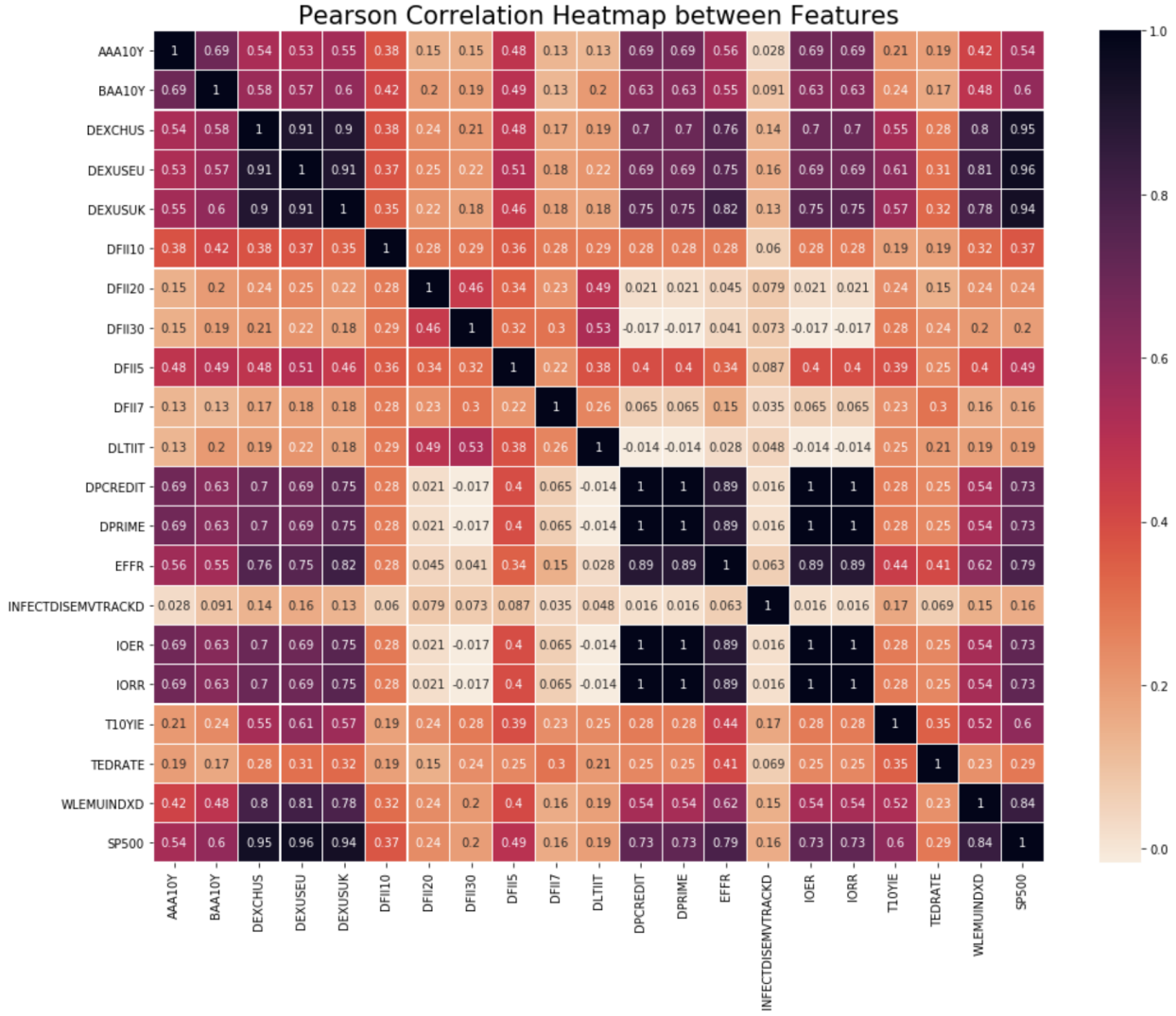

Figure 1.1

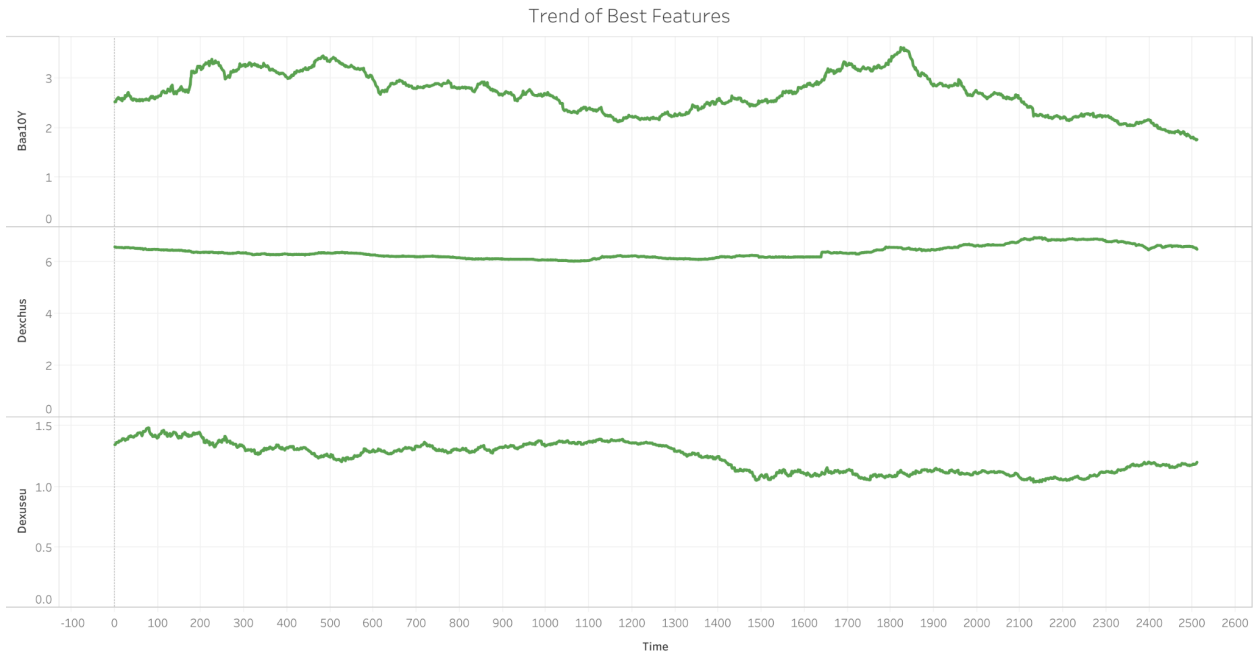

Figure 1.2: Interactive visualization here (Trend of the Features We Selected)

Algorithms

- **ARIMA :** The ARIMA was performed directly on the time series associated with the S&P500. The lag was set to 2 for both the autoregression and moving average and the number of differences taken was set to 1 to achieve weak stationarity, based on the auto.arima function in the forecast R library. Then, using this ARIMA model, we built a forecast to look at trends after 7 days, incorporating a 95% confidence interval, as ARIMA is the most accurate for close projections. We hypothesize that ARIMA will perform best when predicting values on a yearly/monthly basis, because it performs best when given more data.

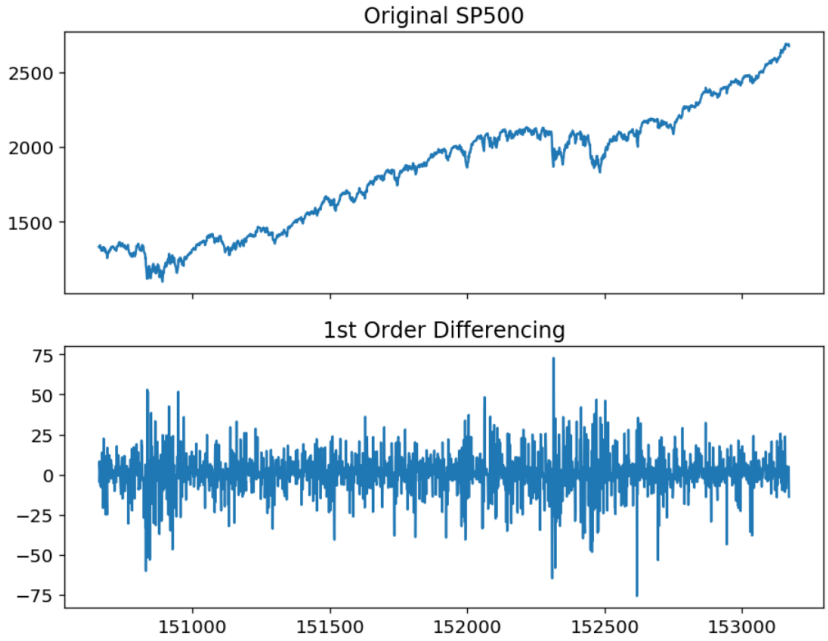

Figure 1.3: Effects of Differencing

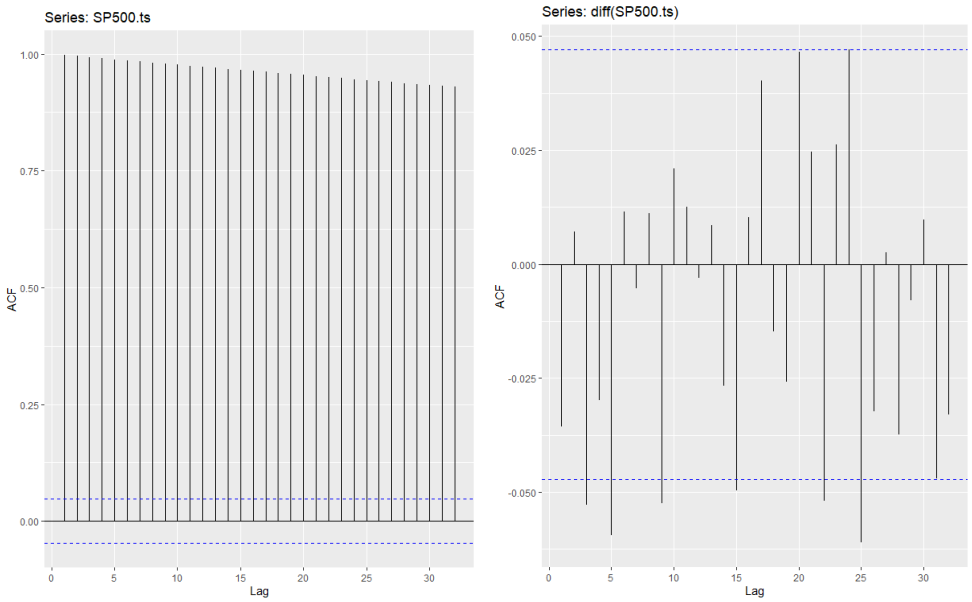

Figure 1.4 Autocorrelation plots before and after differencing

- Effect of differencing can be seen in Figures 1.3 and 1.4, with the trend and the correlation being mostly removed.

- **Random Forests:** We chose random forests because of its historically high scores in the face of all kinds of data. We built a GridSearch model to find the best parameters, considering a max of n_features at every split using 200 trees. We then fit the model on the training data available to us. We hypothesize that our Random Forest model will be better at predicting values on a weekly basis, because RF is able to perform well with smaller datasets.

Scoring Metrics

After splitting the data into 30 splits for walk forward validation, we used the following scoring metrics to base our best parameters on:

- R^2 score : We used R^2 for our random forest regressor to see how close our line of regression was to the true values.
- RMSE: We used RMSE to find out explicitly how our predicted values deviate from the true values.

Results

Outcome (ARIMA)

The results of the ARIMA model, for the 7 day forecast, can be seen in this table and plot:

Date	Lower Confidence Interval Series	Forecast	Upper Confidence Interval Series
1/1/2018	2651.967382	2674.269199	2696.571016
1/2/2018	2642.738954	2674.705476	2706.671998
1/3/2018	2635.158871	2675.031341	2714.903811
1/4/2018	2628.682478	2675.414687	2722.146895
1/5/2018	2623.312690	2675.949959	2728.587228
1/8/2018	2618.911192	2676.606297	2734.301402
1/9/2018	2615.132714	2677.273888	2739.415061

Figure 2.1: 7 Day Forecast ARIMA

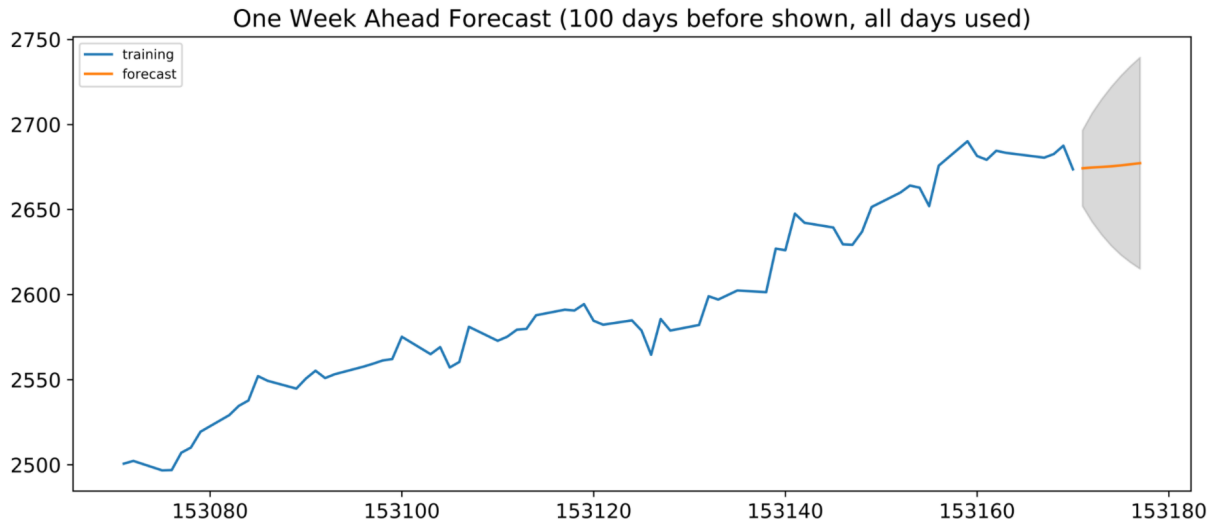

Figure 2.2

Below are the plots for the less reliable month and year long ARIMA forecasts as well:

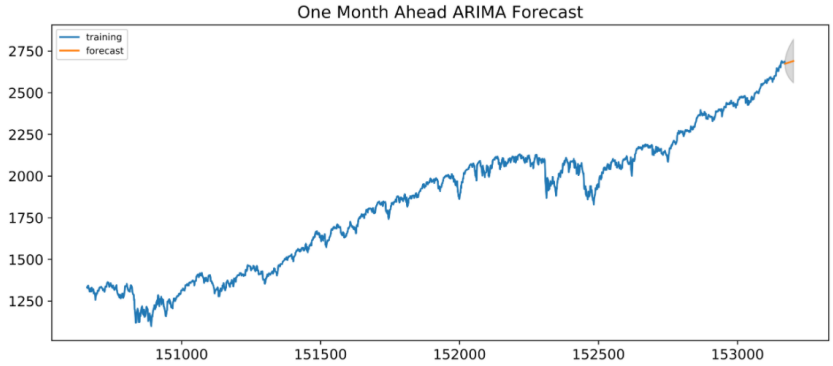

Figure 2.3

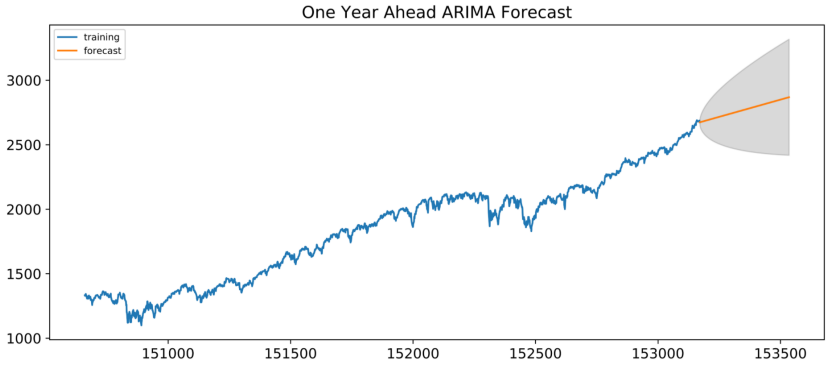

Figure 2.4

Outcome (Random Forest)

As expected, our results with Random Forests were highly accurate when scored on the training set. Figure 3.1 illustrates predicted values as compared to the true values. We were able to achieve an R^2 score of 0.99 and RMSE of 7.4.

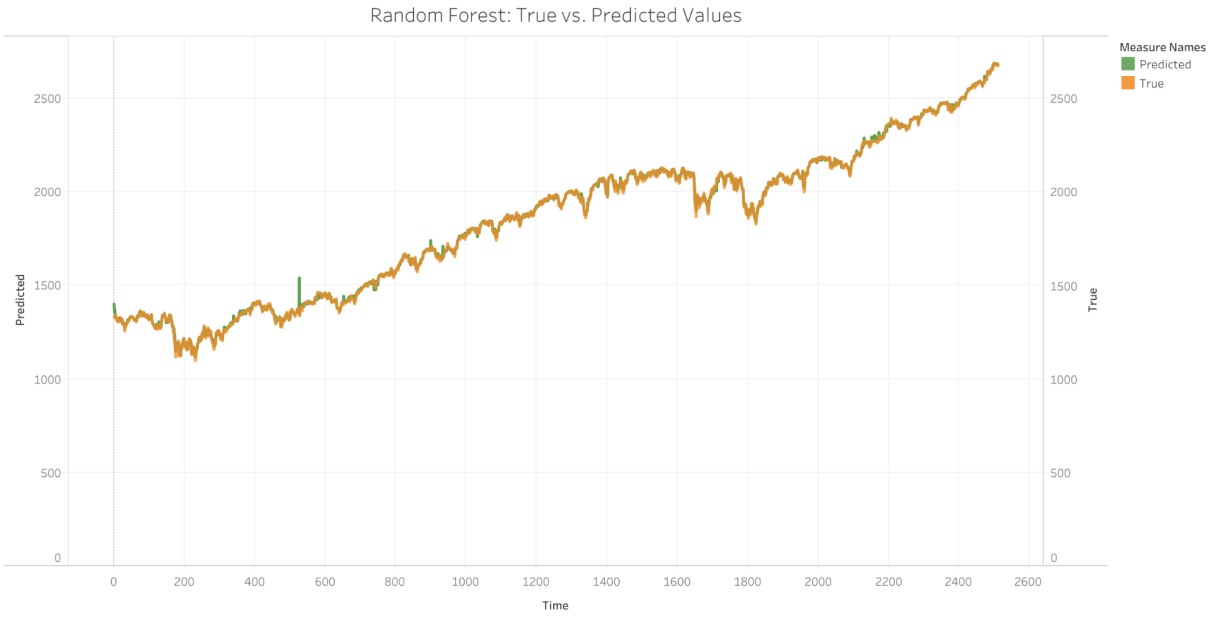

Figure 3.1: Interactive visualization here *(RF True vs. Predicted values)*

Discussion

Given that our datum is of size approximately 2k, an RMSE of 7.5 for the Random Forest is appropriate (considering that the training data is being scored here, not the testing data).

We predicted the values for S&P500 for each specified date, using Random Forest for the weekly predictions and ARIMA for the monthly and yearly predictions. Although ARIMA is best for short-term forecasting, we found that with weekly data we weren't able to run ARIMA because there was too little data for the algorithm to train on.

Both our models worked well and although we didn't have time to score them, our predicted values were close to the true values. The model that we found to work the best was ARIMA using data from previous year, which makes sense because ARIMA is widely used for predicting stock market trends and it performs the best when trained on more data.

True Values:

- True July 12 Value: 3013.77
- True July 12 Year Value: 3013.77
- True March 6 Value: 2972

Weekly Predictions using Random Forest:

- RF Predicted March 6 Week Prediction: 3017.686
- RF predicted Jan 10 Week Prediction: 3263.124249999989

- RF predicted July 12 Week Prediction: 3006.637000000005

Monthly Predictions using ARIMA:

- ARIMA predicted Mar 6 Monthly: 2920.907643
- ARIMA predict Jul 12 Monthly: 3016.616802
- ARIMA predicted Jan 10 Monthly: 3276.426194

Yearly Predictions using ARIMA:

- ARIMA predicted Mar 6 Yearly: 2969.595717
- ARIMA predicted Jan 10 Yearly: 3270.809643
- ARIMA predicted July 12 Yearly: 2992.094325000004

References

https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

https://towardsdatascience.com/time-series-forecasting-arima-models-7f221e9eee06#:~:text=Differencing%20is%20a%20method%20of,used%20in%20an%20ARIMA%20model.

https://scikit-learn.org/stable/modules/model_evaluation.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html