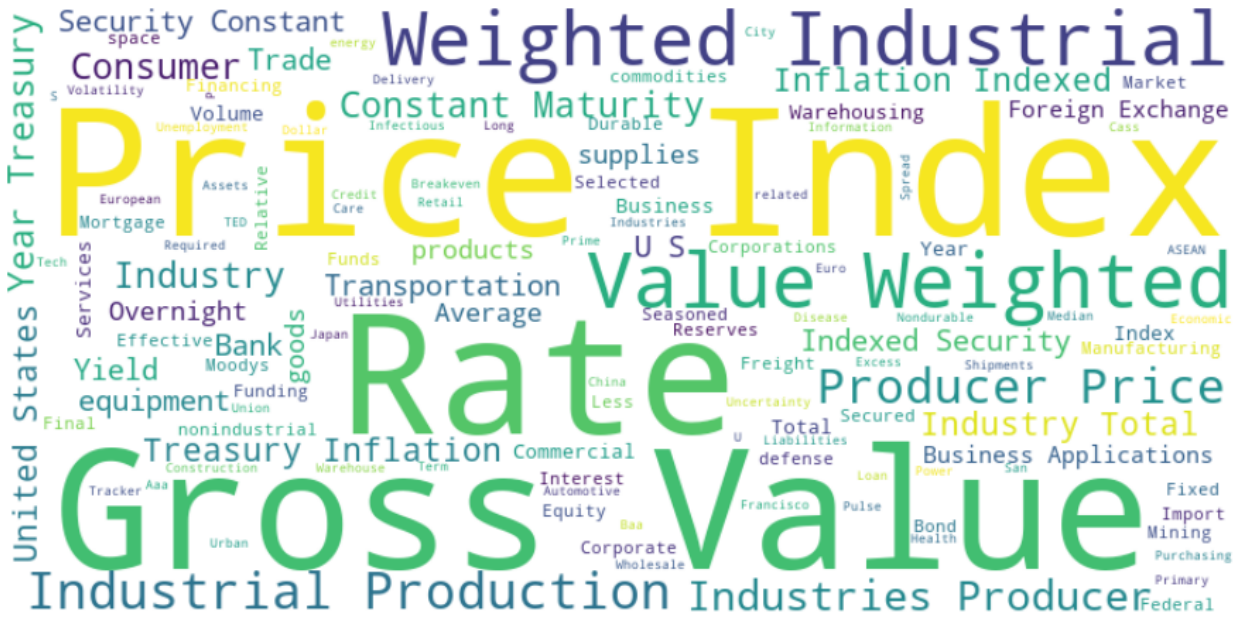


[1]

Predictive Analysis on SP500 Index

04/11/2021

Zirui “Colin” Wang, Yung-Chieh “Jerry” Chan, Jia “Elvis” Shi
University of California, San Diego
{ziw029, ychan, jis283}@ucsd.edu

[1] **The above word cloud was generated based on the lemmatized and stemmed words aggregated from names of economic indicators provided as part of the dataset with NLP techniques.**

Abstract

In this DataHacks competition, we were given the opportunity to explore multiple economic indicators relevant to the US economy, and make predictive models on the future trends (specifically, 2018 - 2021) of the S&P 500. S&P 500 is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the United States. Our goal is to clean, pre-processing, visualize the data, create a Machine Learning model that predicts the weekly, monthly, and yearly trend of the S&P 500 index, analyze and discuss the results based on our model's outputs.

Note

We believe that visualization is an important part of the report, especially given the characteristics of our dataset and our tasks. Therefore, instead of creating a separate section for visualization, we have integrated all visualizations into our report. You will see appropriate visualizations across different parts of this report whenever they are needed. Most visualizations were created with Tableau as well as some Python libraries (matplotlib, seaborn, altair, etc)

Introduction to the Dataset

In this competition, we were given three separate files to analyze and build a predictive model upon. The following describes each dataset file:

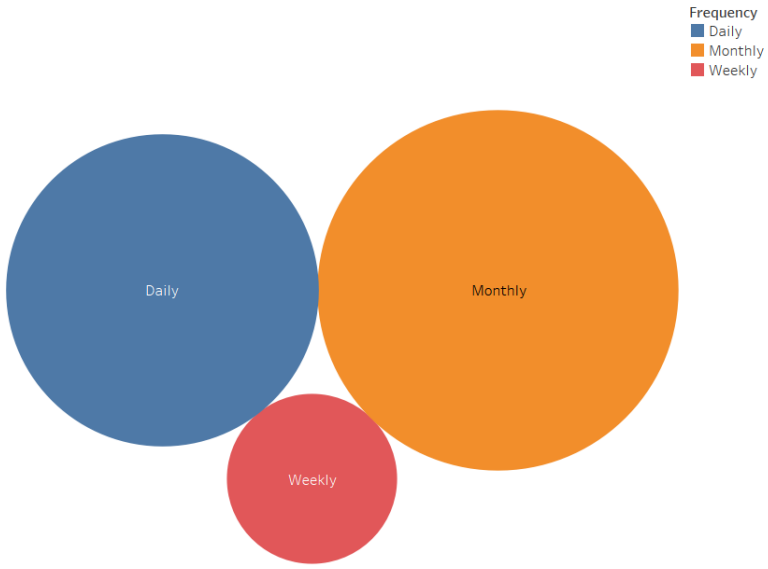
- observations_train.csv: This dataset contains daily/weekly/monthly value of all economic indicators and S&P 500 index starting as early as 1/1/2000 and ending at 12/31/2017. The data for S&P 500, our target for making predictions, started in 2011.
- observations_test.csv: This dataset contains daily/weekly/monthly value of all economic indicators and S&P 500 index starting as early as 1/1/2018 and ending at 2021. The data for S&P 500, our target for making predictions, started in 2018.
- series.csv: This dataset contains helpful information to explain each economic indicator in the observations datasets. It includes the full name of each indicator abbreviation, the rate at which data is collected for each indicator, the units for each indicator's values, whether each indicator has been through seasonal adjustments, as well as a brief description of each economic indicator to concrete our knowledge in these indicators.

Data Cleaning and Preprocessing

Since our data belongs to a time series type, there were inevitably indicators that contain missing values at some point (i.e. a holiday) or have different recording frequencies. Furthermore, the units for each indicator is not necessarily the same, which can make our model more difficult to train without any normalization. Therefore, we did the following steps to ensure that the dataset will be suitable for training the model.

Breaking Down of Dataset

We break down our dataset into daily/weekly/monthly subsets. We'll be mainly using daily subset to make predictive analysis for high-resolution needs (daily/weekly predictions), daily/weekly subset to make predictive analysis for medium-resolution needs (weekly/monthly predictions), weekly/monthly subset to make predictive analysis for low-resolution needs (monthly/quarterly/yearly predictions). The below figure shows an approximate proportion indicators that have daily/weekly/monthly rates:



Data Dropping & Interpolation

For daily subsets, we removed all entries recorded at weekends, since many indicators do not have records on weekends, and the stock market doesn't open during weekends. For missing values in each subset, we interpolate and impute the missing value based on the moving average of that indicator with respect to time.

Normalization

We performed group-wise normalization of data where a group represents one economic indicator using z-score standardization.

Feature Engineering

Adjust Frequency

First step of Feature Engineering is to adjust all series' frequency to daily. Since some series are collected at a monthly or a weekly rate, We filled the missing values with the last recording that comes before it. For example, unemployment rate (UNRATE) is collected on the first day of each month. Suppose the normalized unemployment rate is 0.5 at April first. We will set the normalized unemployment rate of every day in April to 0.5. There are many interpretation methods that can be used in this step to make the series more smooth and reasonable. However, most of those interpolation methods involve the data that couldn't be collected at the time, for example, the unemployment rate of next month.

Growth

The first feature we are going to add to our dataset is the difference between a series's record and it's previous record. We calculated the differences for all series, except for SP500 and append it to our dataset. For monthly and weekly data, we filled the uncollected value with the difference between the latest record and the record before that. For example, the "growth" of unemployment rate in April fifteenth will be filled by the difference between April unemployment rate and March unemployment rate since there is no way to acquire the actual value.

Fourier Transform Extrapolation

Fourier transform is a mathematical transformation that decomposes a function to sum of simple waves. Those simple waves can be transformed back to the original function. We use this technique to predict a SP500 value with the past 1000 days SP500 value and add the prediction as a new feature. The extrapolation is completed with the following steps:

1. Take the past 1000 days SP500 value (not including the current value) and use fourier transformation to decompose it to simple waves
2. Keep the top-k simple waves with the lowest frequency
3. Reconstruct the function with those k simple waves by inverse fourier transform
4. calculate the value of the current SP500 value with the new function

By adjusting the value of k, we can control how detailed the reconstructed function will be. If k is large, it'll reconstruct the function using many high frequency simple waves. This results in noisy predictions and the extrapolation will likely to focus on short-term patterns.

In contrast, if we select a small k, the reconstructed function will be more smooth and focused on long-term trends. We decided to use different values of k (3, 5, 10, 100) and add all the extrapolated values as new features.

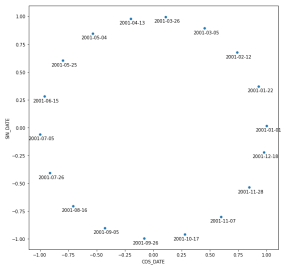
Technical Analysis

We add the following technical analysis index as new features:

1. ma7: 7 day moving average, average of SP500 of the last 7 days.
2. ma21: 21 day moving average, average of SP500 of the last 21 days.
3. 12ema: 12 day exponential moving average, a weighted average of SP500 of the last 12 days which put higher weight on more recent values.
4. 26ema: 26 day exponential moving average, similar as 12 ema but with SP500 of the last 26 days
5. MACD: Moving average convergence divergence, the difference between 12ema and 26ema.
6. Bollinger Band: An interval between ma21 plus / minus two 20 day moving standard deviations.

Cylindrical Time Encoding

The way human recorded date causes some problem to the model:



1. If we pass a date that isn't in the training set. It's completely new to the model and it might not know how to handle it properly
2. If we use month or the day of the year, 12-31 and 01-01 has the largest distance between all pairs of dates but they are only one day apart!
3. If we only use the year, it fails to remain the information of ..

Therefore, we encode the date with cyclical features encoding. First, we convert the date to day of year and project it on a unit circle. The first day of the year, January first, will be on (0,1), January second, will be on (0.01725, 0.9998), and so on. Then we take the x, y coordinates as embedding. This encoding solves the problem mentioned above and maintain some properties of date:

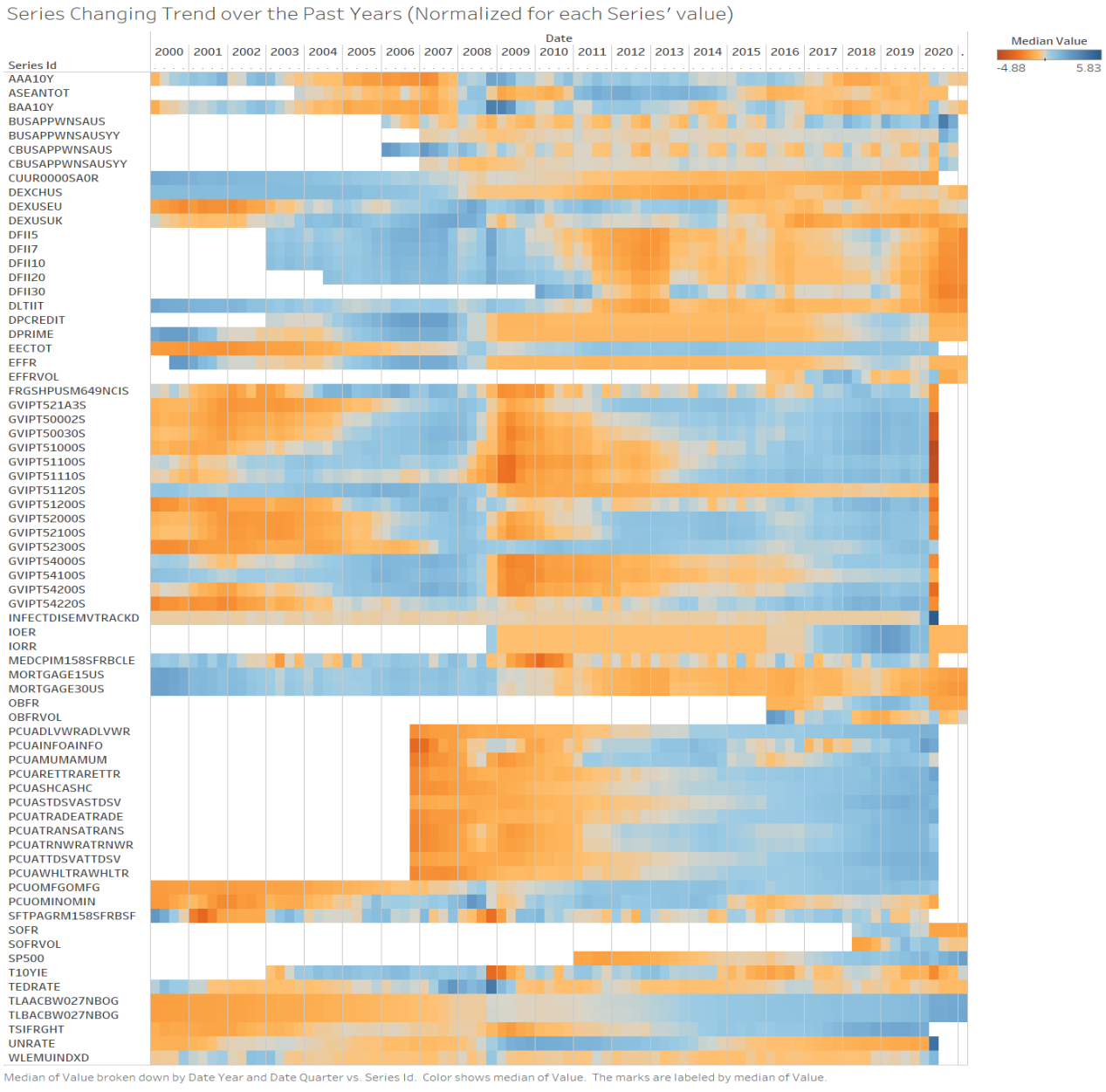
1. Dates that are furthest away from each other, such as January and July, will be one the opposite of the circle.
2. The four quarters of a fiscal year are divided into four quadrants on the cartesian coordinate. The first quadrant represents Q1, the second quadrant represents Q2, and so on. We can tell which quarter a date is in by the sign of the encoding. It might be helpful for the model to discover any seasonal trend.
3. The period of the encoding matches the period of a year. It's easier for the model to discover annual patterns.

Analysis and Modeling

Exploratory Data Analysis

Data Distribution & Separated Trend

First of all, we used a heat map to show the distribution of our normalized data (combining training and testing set) in terms of time and used color to encode their values to show the trend of different economic indicators.

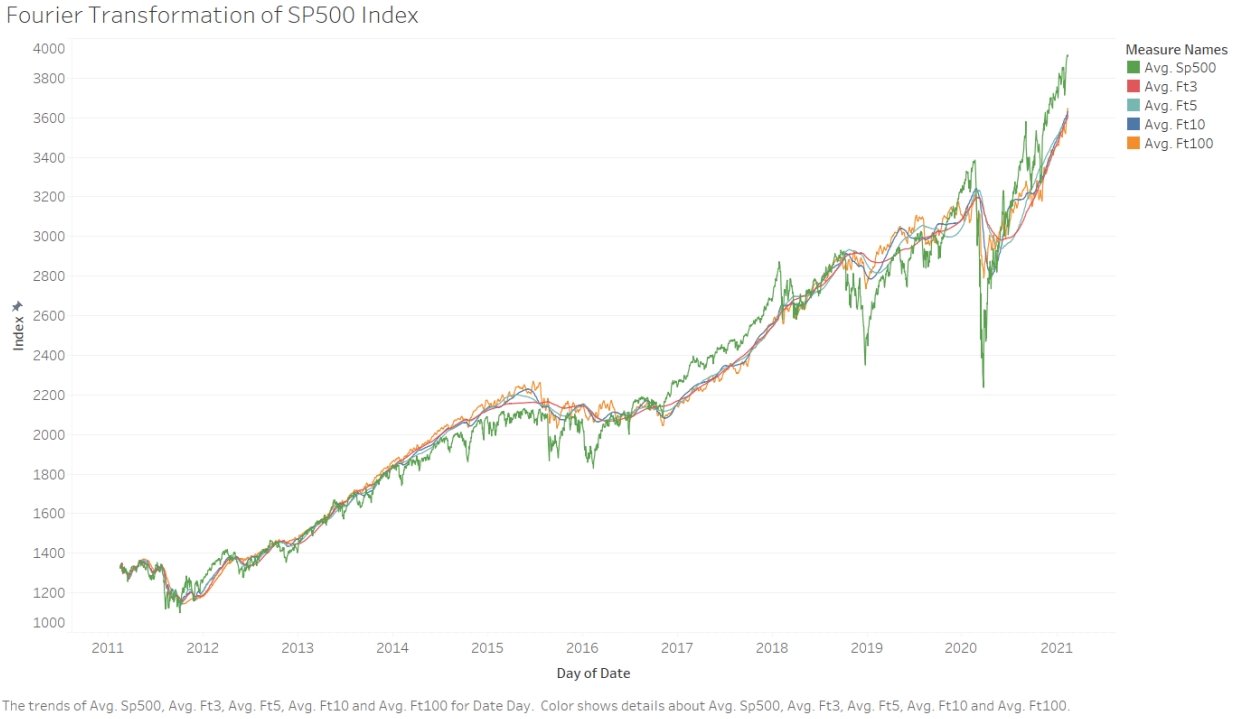


From the above figure, we can see that economic indicators from our dataset start at different times, where the majority of them start at 2000 and end in mid-2020. There are economic indicators started at the middle such as the PCUA~ series, DFII~ series, and BCU~ series. Finally, there are some indicators that do not appear in the training set but appear in the test set, such as the SOFR~ series. Since the S&P 500 data appeared in 2011, most economic indicators are still useful even with a large time-series window such as one year (i.e. we have economic indicator data after 2010 for most series).

In terms of trend, we found three characteristics:

- Indicators that gradually have increment of values
    - Examples: PCUA~ series, TLAA~, TLBA~ series
    - Our target S&P 500 follows this trend as well
- Indicators that gradually have decrement of values
    - Examples: DFII~ series, MORTAGE~ series
- Indicators that have cyclic trend of values
    - Examples: BUS~ series, CBUS~ series, MED~ series
- Indicators that have increment throughout, but experienced a major drop of value in year 2009
    - Examples: GVIPT~ series

Fourier Transform of S&P 500



The above figure shows the trend of S&P 500 over the years and some trends after performing Fourier Transformation. We used different top-k simple waves (i.e. 3, 5, 10, 100) to selectively keep important information in the trend of the S&P 500 curve to smooth our data. By doing this, we avoided some unnecessary noise that can potentially cause negative effects in making long-term predictions. Simply speaking, the smaller the k, the smoother the curve.

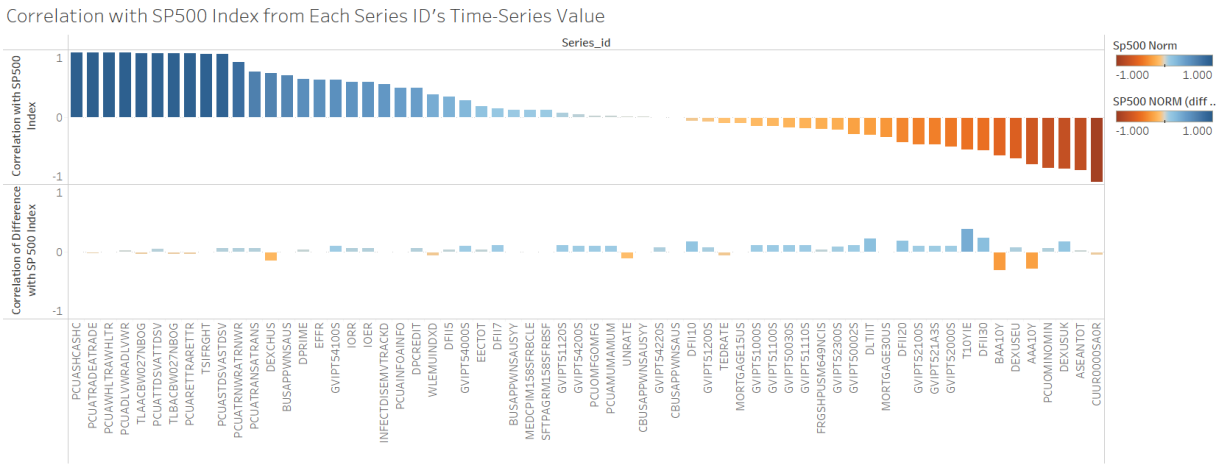
Technical Indicators of S&P 500

The trends of Avg. EMA, Avg. MA21, Avg. MA7, Avg. MACD, Avg. Momentum, Avg. MACD and Avg. Momentum for Date Day. Color shows details about Avg. EMA, Avg. MA21, Avg. MA7, Avg. MACD and Avg. Momentum.

In our feature engineering procedure, we also derived some technical indicators of S&P 500 data over the years. These indicators mostly cover the moving average of S&P 500 data as well as MACD and Momentum. As these indicators are important for many stock traders for making decisions, we believe that these indicators can also be helpful for our ML model to make predictive decisions.

Now, after we have derived some additional features on fourier transformation and technical analysis, let's analyze the correlation between these indicators and S&P 500 in terms of trend.

Correlation & Difference Correlation & TLCC between S&P 500 and Other Economic Indicators

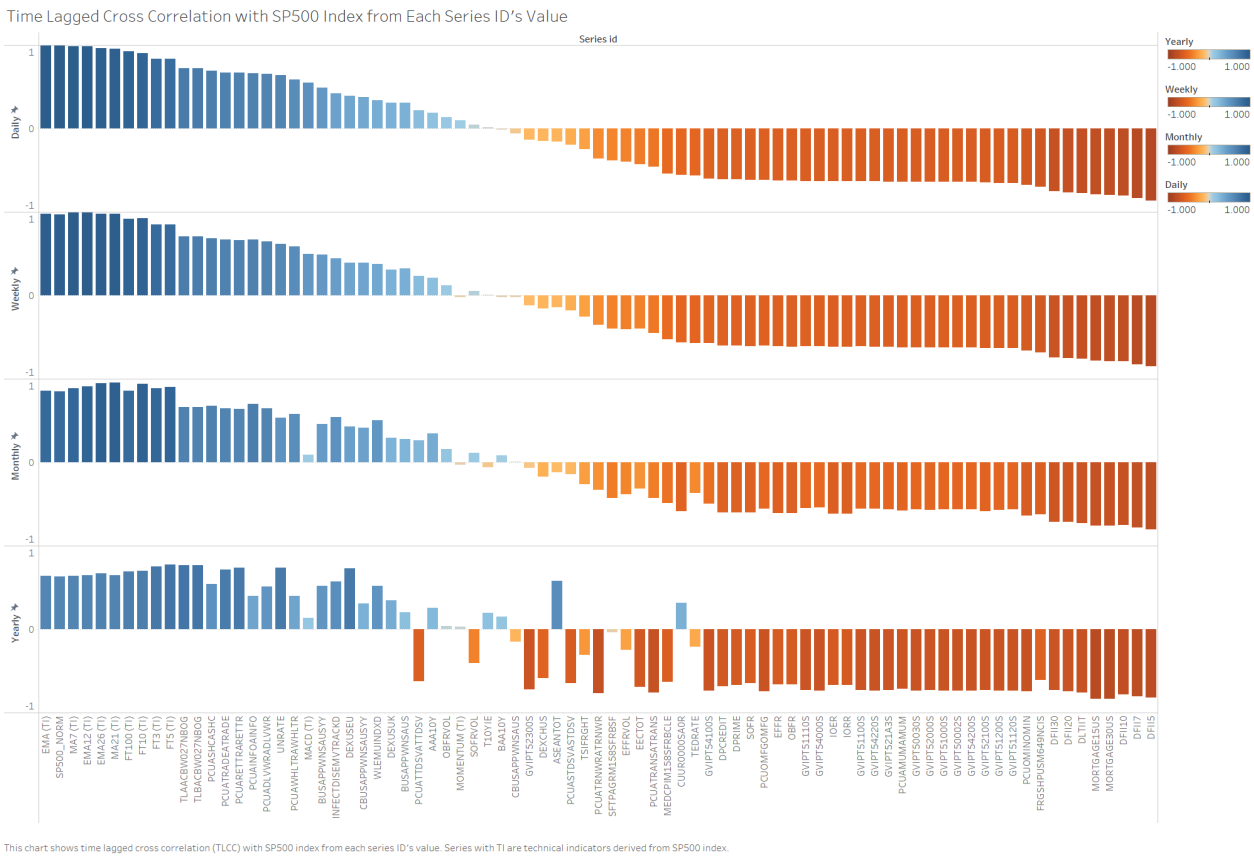


This chart shows correlation between each series ID and SP500 in terms of direct correlation and difference (between each day/week/month interval) correlation.

In the first figure, we compared the correlation between S&P 500 and other economic indicators in terms of direct correlation and the correlation of difference (between nearest time gap). The upper chart shows direct correlation and the lower chart shows correlation of difference. We can see that the trend for S&P 500 is mostly correlated with PCUA~ series trend, which is consistent with our conclusion derived from the heatmap above. This makes sense since this series represent Producer Price Index by Industry, and studies have shown that Producer Price Index has a positive relationship with S&P 500[2]. On the other hand, CUUR~ series is mostly negatively correlated with S&P 500 index. This also makes sense since CUUR~ represents Consumer Price Index, which can be affected by PPI. Therefore, as PPI increases as S&P 500 increases, the CPI will decrease as a result, therefore displaying a negative correlation with S&P 500 Index.

If we look at the lower figure that compares the correlation of difference, things turned out differently. While the direct correlation may reflect a consistency in a long-term perspective, the correlation of difference better represents the casual relationship - that is, if a difference in price between two consecutive dates is seen in some economic indicators, will we see the same difference (whether positively or negatively) in the S&P 500 index? Here, we see that in fact T10YIE has the strongest positive correlation of difference. T10YIE represents the 10-Year Breakeven Inflation Rate. This is an important indicator for many investors and institutions to decide where to invest the money, so their actions taken from T10YIE also affects the trend of S&P 500. On the other hand, BAA10Y and AAA10Y have the strongest negative correlation of difference. The reason is similar to T10YIE, where they represent 10-Year Treasury Constant Maturity, which is also an important indicator for investors and institutions to decide where to invest the money.

[2] https://academicarchive.snhu.edu/handle/10474/1666



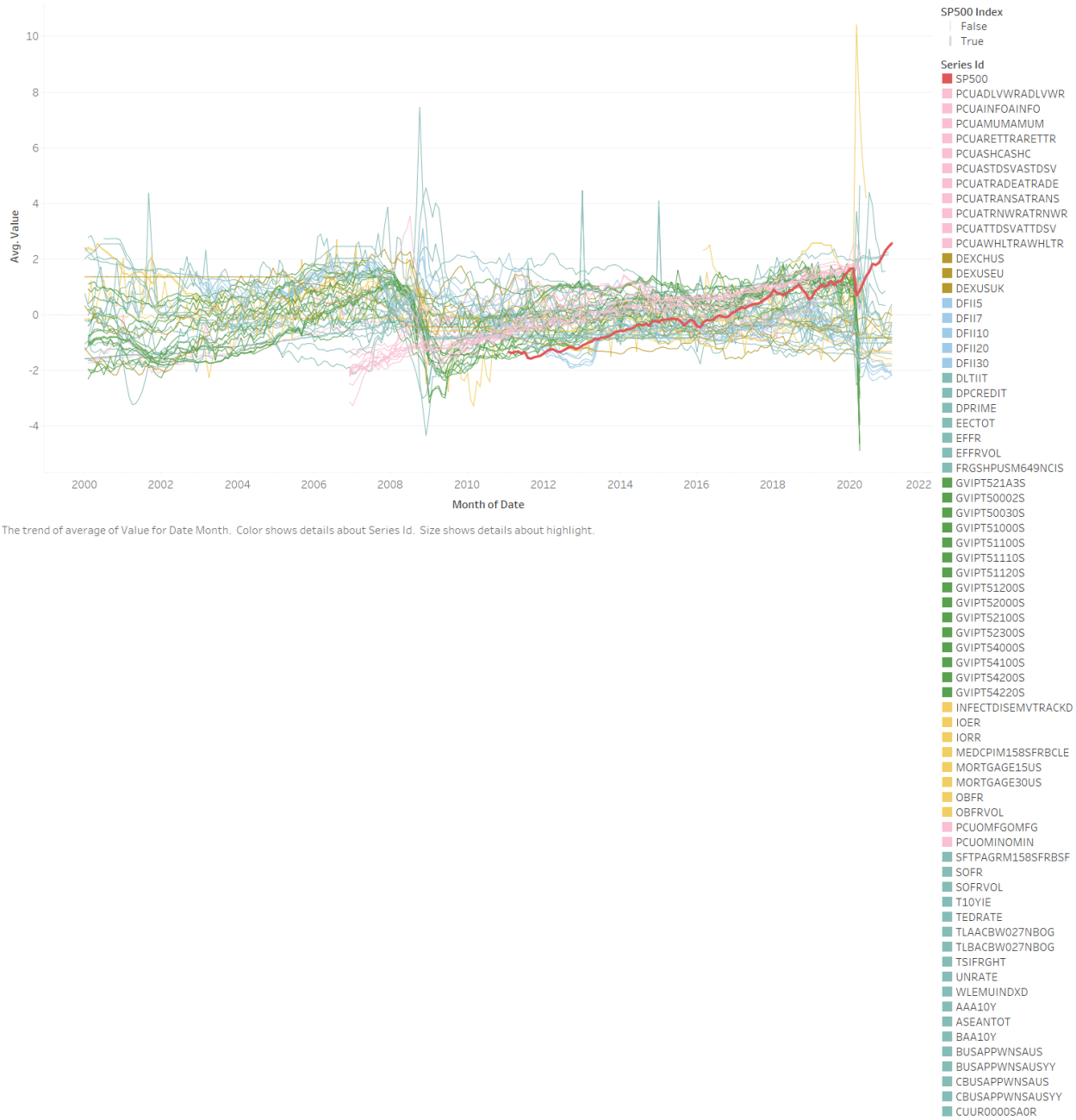
This chart shows time lagged cross correlation (TLCC) with SP500 index from each series ID's value. Series with TI are technical indicators derived from SP500 index.

This figure shows a time lagged cross correlation between S&P 500 index and other economic indicators. For this figure, we also included technical indicators and fourier transformation results from our derived features. We did so because we are interested in the time lagged cross correlation, which prevents data leakage if we used previous technical indicator trends and fourier transformations to predict the future of S&P 500 index.

We can see that while the general time lagged cross correlation is similar when the lag is on a daily/weekly/monthly basis, this correlation shows a reverse for some indicators when the lag is on a yearly basis. Some significant correlation changes include ASEANTOT (Import Price Index) and PCUATTDSVATTDSV (we believe that this is an exception since we didn't observe this lagged difference in the above heatmap).

On the other hand, we're glad that many of our derived features actually have strong positive correlations with the trend compared to the actual S&P 500 index even with a long time lag, such as the various yearly moving average indicators. However, we're suspicious of the effectiveness of it since the S&P 500 index seems to be monotonically increasing over the years, so their technical indicators, especially the moving average, also go up monotonically. This will be much more convincing if S&P 500 index shows a cyclical pattern, which doesn't present in the given datasets.

EDA Appendix: Line Chart Trend for S&P 500 in Comparison with Other Economic Indicators



The trend of average of Value for Date Month. Color shows details about Series Id. Size shows details about highlight.

The above figure serves a supporting line chart that shows the trend of S&P 500 over the years compared to the trend of other economic indicators in the dataset. Since they are mostly analyzed in previous analysis, this chart provides a more general view of the visualization of our dataset. Note that indicators belong to the same or similar categories are encoded with the same color.

Modeling

In this problem, since we are dealing with time series data, we decide to implement an RNN-based deep learning model for its ability to remember information from the past. In stock market prediction, the given state of a stock was not only determined by the most latest information, but also included the information of a similar pattern that had happened hundreds of steps before . That is why the memory cells of time series deep learning modeling are extremely useful for stock price prediction.

First, we design the vanilla LSTM as the baseline model. It's a model with one single LSTM model layer. Then, we also created another model for comparison.

Stacked LSTM is a model with the stack of N different LSTM cells, in this case we use N=3.

Conv-LSTM is a good model to compare with the baseline. CNN had often been used in images or spatial data, which assume that information that is closer to each other are more related to each other, we assume this is also true for time-series data.

We didn't include Bidirectional LSTM because we don't have access to future stock information.

Also we had tried the stacked-GRU model, **because it makes direct predictions based on all hidden stages without gate control. Also, with less data, GRU may have better performance because it has fewer parameters compared to LSTM, thus it needs less data to generalize.**

Meanwhile, beside time-series based modeling, we also used data from other series on the same date as the S&P 500 prediction to do a regression based on the machine learning model. We had tried XGboost and LGBM for this experiment.

Proposal and Experimental Testing

Proposal

Our hypothesis is that the model prediction will be more accurate when the time constraints are given for longer periods. That is, a prediction made by looking at the data for the entire past year will be more accurate than looking at the data for the path month, and finally past week. We'll see if our prediction agrees with this hypothesis or not in the following experiments.

The reasoning behind why we designed the model in such a way is described in the Modeling subsection in Analysis and Modeling section.

Experiment One

Since the stock market is highly volatile, thus it's very difficult to use time series data based on historical data, thus we have to use the same date stock information to predict the SP500 information without the use of historical data.

We have tried boosting regression models including XGboost and LGBM to do this regression.

However, the results don't look very good because the feature prediction model didn't capture the rising trend of the SP500 through time.

Experiment Two

Use previous stock data to predict the change in the next day of SP500 (Normalized Data):

Model Name	Training Loss	Validation Loss
Stacked GRU	5.5260e-04	41.088
Stacked LSTM	5.3115e-04	42.985
ARIMA	5.2239e-04	40.512

We find out that the change of SP500 in continuous two day is very close to each other after normalization, thus our deep learning result always has a prediction very close to zero.

Prediction Results

Date	Interval	Ground Truth	ARIMA Predictions	GRU Predictions	Final Prediction	Absolute Difference
1/10/2020	7	3265.35	3277.425	3277.079	3277.079	11.729
1/10/2020	30	3265.35	3263.298	3273.997	3263.298	2.052
1/10/2020	365	3265.35	3274.588	3276.196	3274.588	9.238
3/3/2020	7	2972.37	3126.567	3029.073	3029.073	56.703
3/3/2020	30	2972.37	3019.358	3089.737	3019.358	46.988
3/3/2020	365	2972.37	3032.085	3092.695	3032.085	59.715
7/12/2019	7	3013.77	2989.185	3003.474	3003.474	10.296
7/12/2019	30	3013.77	3005.945	2999.52	3005.945	7.825
7/12/2019	365	3013.77	3001.357	3003.017	3001.357	12.413

Discussion

With the use of both statistical Machine Learning model (ARIMA) and a Deep Learning framework (GRU), we decided that our final prediction will be based on GRU results for weekly intervals, and ARIMA results for monthly and yearly intervals.

In the result, we discovered that one-week data worked best with GRU, but with the increase of historical data data, the ARIMA modeling actually worked better. One hypothesis is the deep learning model like GRU will have much higher number of parameter compared to ARIMA, thus it will overfit the dataset once the number of data get greater, however, when the number of training data is less, the model of deep learning will have less chance of overfitting and thus outperform the ARIMA machine learning model.

Our results disagree with the hypothesis that a longer interval for the model to make predictions upon will make these predictions closer to the ground truth. We believe that it's caused by the fact that the stock market is highly volatile and a longer interval to make predictions upon will lead to some unwanted noises, while a shorted interval like a week will not give our model enough context to make accurate predictions. Therefore, using a monthly window gives us the best results.

Conclusion

In this competition, we analyzed the S&P 500 index data across the years in comparison to other economic indicators. We cleaned and normalized the data, derived additional features and indicators through feature engineering. We performed EDA and visualizations on our datasets and derived insights on which indicators might be helpful for what kinds of predictions. We built the model with various frameworks including different LSTM structures and XGBoost. Finally, we used our best model to make predictions based on the specified requirements on the prompt. Although the stock market is highly volatile and can be heavily influenced by external factors such as politics, natural disasters, environmental factors, our model still learns to make a decent prediction based on many economic indicators from different perspectives and technical indicators from the past of the S&P 500 index. We hope that our model can help traders and investors better understand why S&P 500 is moving in a certain direction in terms of some predictive trends from the past!