Math 156 Final Project Report

Sophia Yang, 505097202
Nora Liu, 705346012
Yixin Wan, 905126641

0. Contributions
Literature Search: Sophia Yang, Nora Liu, Yixin Wan
Dataset: Nora Liu(data extraction), Yixin Wan(data cleaning)
Coding: Sophia Yang(Logistic Regression), Nora Liu(Time Series), Yixin Wan(Support Vector Machine)
Writing: Sophia Yang, Nora Liu, Yixin Wan

I. Introduction

Financial indices such as Nasdaq Composite and S&P 500 are widely accepted to be important indicators of the entire financial market in the United States. A large number of previous studies have been done in the search for a prediction mechanism for the directional movement of financial index prices. However, trends of financial index prices are often hard to predict due to uncertain noises, both from the company's operating itself and the overall performance of the financial market. Due to the existence of such noises, existing models in the field of econometrics- such as time series analysis- fail to achieve satisfactory predictive accuracy. Inspired by the rising number of studies on applications of Machine Learning methods for price prediction tasks in the stock market, our group project aims to examine two different types of Machine Learning models' ability in predicting trends of financial indices prices. Specifically, we choose to evaluate the prediction performance of two Machine Learning models, one each from the two categories of models that we have discussed in class- generative and discriminative models. Our project can be roughly split into two parts: first, we compare the prediction accuracy of the traditional time series model in the field of econometrics and the two Machine Learning models. Then, we further establish a fair comparison between the generative and discriminative Machine Learning models in an attempt to find the best choice for predicting the directional movement of financial indices.

II. Datasets

For the first part of our project, we want to establish a comparison between the time series model and the two machine learning models. Since the time series model only utilizes historical prices to forecast future values, in an attempt to make a fair comparison between the models, we choose to use historical prices of the financial indices Nasdaq Composite and S&P 500 to construct our training and testing dataset. We made use of the yahoo_fin package for Python (https://pypi.org/project/yahoo-fin/) to retrieve daily prices of the two financial indices for six years (March 2016 - October 2021) as our base dataset. Utilizing the base dataset, we constructed and conducted experiments on different sizes of the training dataset (i.e. how long the period we should choose to extract the financial indices from as our training set) ranging from 1 (March 2020 - March 2021) to 5 years (March 2016 - March 2021) of the financial index data, and constructed a 90-day test dataset from prices of the stock indices from April 2021 to October 2021. Specifically, since the time series model makes predictions of price trends utilizing historical data (for instance, the index prices of the prior two days), we also choose to use historical data as the features for each input vector in the training and testing dataset for two machine learning models.

For the second part of our project, we choose to experiment with sizes of training datasets from 3 to 5 years, since we observe from outcomes of the first part of our research that a larger size of training set would help the models achieve the most stable prediction accuracy. We choose the optimal number of historical prices to include as input features in our further experiments to include based on results from the first part. To better compare the two machine learning methods' ability to forecast the directional movement of financial index prices, we also choose to additionally incorporate the historical daily stock prices of Google, Apple, and Tesla as features in the training dataset. This is because we assume that those three stocks are representative of the trend in the stock market, and might play a significant role in forecasting the market's future movements. Besides making comparisons between the two machine learning models, we investigate whether including additional information like the prices of the three stocks would help with improving the prediction accuracy of the models.

III. Methods

Recall that we wish to establish a comparison between the traditional time series model and the two types of machine learning models. As for specific model choices, we choose to use the Autoregressive integrated moving average(ARIMA) model to represent the time series method, Support Vector Machine(SVM) to represent the generative machine learning models, and Logistic Regression(LR) to represent discriminative machine learning models. We evaluate these models' ability to predict financial index price trends by comparing their prediction accuracy for the directional movement of prices. The prediction accuracy, in our case, is defined as the percentage of accurate daily predictions (an upward or downward trend the next day) during a period of time.

Recall from the last section that to establish a fair comparison between the time series model and the two machine learning models, we choose to use historical data as the features for each input vector in the two machine learning models. In other words, taking the Nasdaq composite as an example, the input vector for predicting Nasdaq index price trend on a day t would consist of historical prices from the previous n days, denoted as:

$$X_t = [p_{t-n}, p_{t-n+1}, \ldots, p_{t-1}],$$

where p_i denotes the index price on the day i. The output y_t is then the price trend on day t. Specifically, $y_t = 1$ if the Nasdaq index price increases on day t, and $y_t = 0$ if otherwise. Therefore, the number of historical daily prices that we choose to use, n, acts as a hyper-parameter to control the dimension of input vectors and thus how much previous information we feed into the model. For each of the two machine learning models, we experimented on different values of n, ranging from 1 day to 5 days. In addition, we also conduct experiments on different sizes of training datasets for all three models to observe the influence of training dataset sizes on prediction outcomes. For each of the models, we experiment on training sets of sizes ranging from 1 year to 5 years of daily financial index price. We then evaluate the prediction accuracy of the three models on a test set of size 90 days.

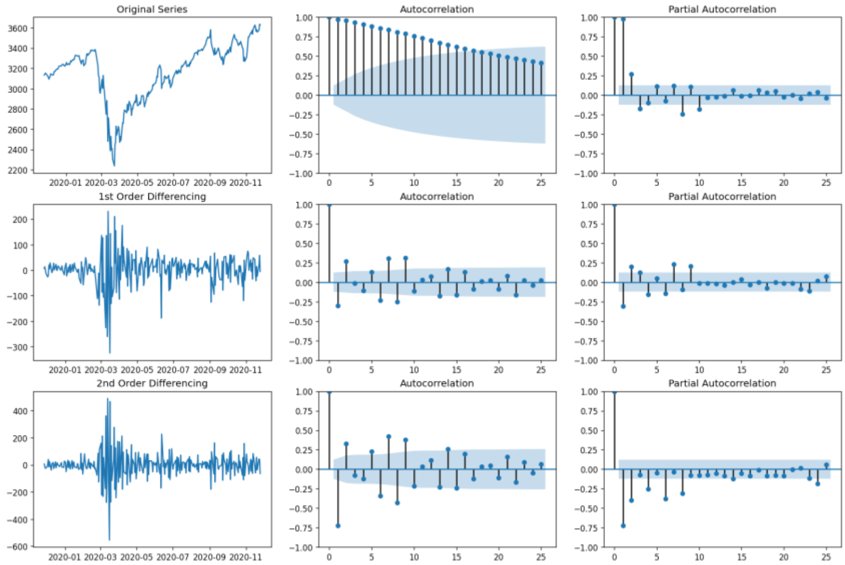
IV. Experiments

i. Time series

Time series forecasting is a model to predict future prices based on historical data, which is what we need in financial index prediction. We are using ARIMA to predict the 90 days financial index trend based on 1 to 5 years of financial trend data.

The first step is to choose parameters for the time series model. The three parameters we need to determine are p (the order of the Auto Regressive (AR) term), q (the order of the Moving Average (MA) term), and d (the order of differencing).

Firstly we need to make our data stationary. The Dicky Fuller Test on the right shows that our financial index data is not stationary with a p-value larger than 0.05. Thus we need some orders of differencing to get nearly stationary data.



On the left, we have the S&P 500 financial index data and its first and second-order differencing results. We can see from the plots that the time series reach roughly stationary at around the first order of differencing, so we take d = 1. Since the datasets look more stationary at d = 2, we will be testing out d = 2 just in case. But primarily we will use d = 1 to avoid over-differencing.

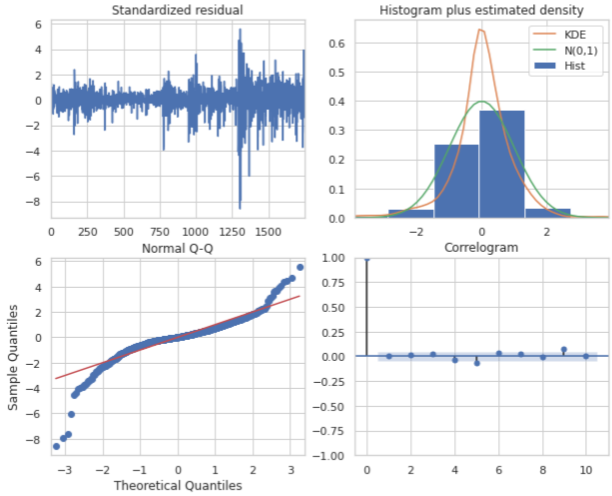
Now we come to the AR and MA terms. At the first order of differencing, the partial auto-correlation graph shows that the lags go inside the significance limits at about 3 to 4 lags, so we take p = 3. The auto-correlation shows that the lags are significant within 3 lags, so we take q = 3.

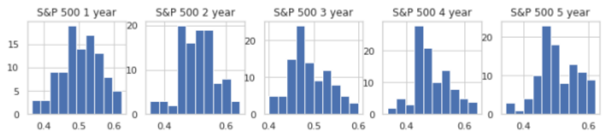
After choosing the parameters, we now come to fit the model.

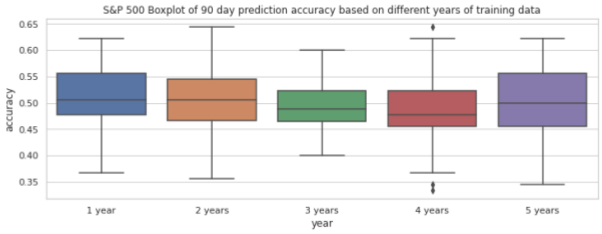
The first thing here is to try auto_arima and test out whether our choice of parameters is the optimal choice. It turns out that (3,1,2) is the optimal choice for NASDAQ and S&P 500 datasets. It makes sense to be conservative and have smaller p and q especially since the last lag is approximately within the significance limit, but we will be testing both models just to make sure.

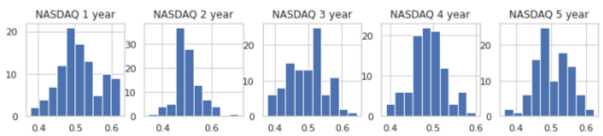
By just looking at the training set, the model can learn the trend pretty well. The residuals of our model are uncorrelated, without an obvious pattern, and roughly normally distributed from the qq-plot and histogram of residuals.

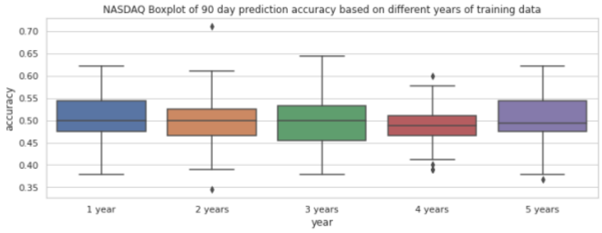
As stated in the previous section, one purpose here is to see whether different time ranges of training set data will affect prediction accuracy. Therefore, 1 to 5 years of training data will be supplied. For each year's range of data, we will be collecting accuracy on 100 different year ranges and taking the average of the accuracy scores. For the ARIMA (3,1,2) model, the 90-day prediction accuracy is roughly 0.5. There is not much difference in prediction accuracy for different years of training data. As we can see from the side-by-side boxplots that the mean accuracy prediction is about the same at 0.5. Even though the p-value from the ANOVA test suggests borderline statistical significance, it might come from the large sample size, and accuracy at this low does not make much difference if we plus or minus 0.01. Thus we can safely say that increasing the training size does not significantly lift the testing accuracy. One good thing is that the testing accuracy roughly follows a normal distribution, without many skewed datasets. This applies to both S&P 500 and NASDAQ.











Test Acc 90 Days	**ARIMA(3,1,2)**		**ARIMA (3,1,3)**		**ARIMA (3,2,2)**	
Training set size	**NASDAQ**	**S&P 500**	**NASDAQ**	**S&P 500**	**NASDAQ**	**S&P 500**
1 year	0.51011111	0.506444	0.4992222222	0.5146666666	0.56011111	0.5394444444444

	11111108	4444444444	2222223	666667	1111111	446
2 year	0.5009999999999998	0.50777777777777777	0.4876543209876543	0.5104444444444447	0.5716666666666667	0.5424444444444447
3 year	0.49533333333333324	0.489777777777777753	0.4869999999999998	0.4969999999999999	0.5498888888888892	0.5521111111111116
4 year	0.48922222222222221	0.48822222222222222	0.4863075196408528	0.5053333333333333	0.5713333333333337	0.569555555555556
5 year	0.50322222222222219	0.50399999999999997	0.5014590347923681	0.517888888888889	0.58722222222222225	0.5691111111111116
P value of ANOVA test	0.06306429010073533	0.034630826310048025	0.05666265436863111	0.04329808179998561	0.000745544778381004 3	1.6467630016891642e-06

As we said earlier we want to see whether there would be any changes if we use order of differencing d = 2 and MA term q = 3. ARIMA (3, 1, 3) model's prediction accuracy is not significantly different from that of ARIMA (3, 1, 2). But by using higher order of differencing, (3, 2, 2), we get statistically significantly higher prediction accuracy in training data sets of all year ranges. These findings apply to both S&P 500 and NASDAQ. This suggests that a higher order of differencing is needed for financial index data.

One-way anova p-valuecompared to ARIMA(3,1,2)	**ARIMA (3,1,3)**		**ARIMA (3,2,2)**	
Training set size	**NASDAQ**	**S&P 500**	**NASDAQ**	**S&P 500**
1 year	0.1454351828048392	0.26921789571408367	2.0601020002409024e-07	0.0001459021831670028
2 year	0.07535705697007163	0.7341590827314604	2.0228388213834886e-15	3.7200545354976486e-05
3 year	0.27587885637252885	0.2917529733540044	5.313532203062428e-10	1.0571462193750493e-21
4 year	0.6532131590717771	0.039949796608052174	5.332608449947042e-22	1.171729477845059e-21
5 year	0.7950562697245482	0.11302297667961585	2.353689114468512e-21	4.2292802361362624e-14

ii. Support Vector Machine

We choose Support Vector Machine (SVM) to represent the category of generative machine learning models that we have learned in class. In theory, SVC can achieve high accuracy especially on two-class classification tasks like in our case because the model is structured to maximize the margin between the decision hyperplane and the support vectors or point in the two classes that are closest to the decision plane. This way, SVC helps us obtain the optimal hyper-plane for the classification decision. Specifically, in our implementation of the SVM model, We choose to use the C-Support Vector Classification (SVC) function provided in the sickit-learn SVM library for Python. We choose to train the model with a Radial Basis Function (RBF) kernel. We adjust the kernel coefficient, gamma, to its optimal value for experiments on each size of datasets. In Sickitlearn's svm.svc library, value of the hyper-parameter C is inversely proportional to strength of regularization with a squared l2 penalty. We choose an optimal value of C equal to 1e03 in all our experiments.

In the first part of our experiments, we used only historical prices of the financial index as input features. We experimented on different training set sizes ranging from 1 year to 5 years, and on different numbers of historical daily prices to include as input features, ranging from 1 day to 5 days. We examine SVM's valid accuracy on both the NASDAQ dataset and the S&P 500 dataset. In the table below, we document the optimal number of daily historical prices that induces best valid accuracy for each training set size.

NASDAQ						S&P 500				
Training Set Size	C	gamma	n	Valid Acc 90 days		Training Set Size	C	gamma	n	Valid Acc 90 days
		1.00E-08	1	0.626373626				1.00E-07	1	0.604395604
		1.00E-08	2	0.648351648				1.00E-07	2	0.626373626
1 year		1.00E-08	3	0.648351648		1 year		1.00E-07	3	0.626373626
		1.00E-08	4	0.659340659				1.00E-07	4	0.615384615
		1.00E-08	5	0.604395604				1.00E-07	5	0.604395604
		1.00E-06	1	0.538461538				1.00E-05	1	0.549450549
		1.00E-06	2	0.582417582				1.00E-05	2	0.538461538
2 years		1.00E-06	3	0.571428571		2 years		1.00E-05	3	0.549450549
		1.00E-06	4	0.637362637				1.00E-05	4	0.571428571
		1.00E-06	5	0.527472527				1.00E-05	5	0.549450549
		1.00E-06	1	0.571428571				1.00E-05	1	0.527472527
		1.00E-06	2	0.571428571				1.00E-05	2	0.549450549
3 years	1.00E+03	1.00E-06	3	0.615384615		3 years	1.00E+03	1.00E-05	3	0.571428571
		1.00E-06	4	0.571428571				1.00E-05	4	0.505494505
		1.00E-06	5	0.549450549				1.00E-05	5	0.516483516
		1.00E-07	1	0.561797753				1.00E-05	1	0.550561798
		1.00E-07	2	0.573033708				1.00E-05	2	0.539325843
4 years		1.00E-07	3	0.573033708		4 years		1.00E-05	3	0.528089888
		1.00E-07	4	0.550561798				1.00E-05	4	0.516853933
		1.00E-07	5	0.539325843				1.00E-05	5	0.505617978
		1.00E-06	1	0.527472527				1.00E-05	1	0.505494505
		1.00E-06	2	0.593406593				1.00E-05	2	0.516483516
5 years		1.00E-06	3	0.703296703		5 years		1.00E-05	3	0.626373626
		1.00E-06	4	0.615384615				1.00E-05	4	0.604395604
		1.00E-06	5	0.549450549				1.00E-05	5	0.450549451

In the second part of our experiments, we choose to use the optimal number of daily historical prices that help the model achieve the best prediction accuracy in the first part of experiments. According to experimental results, this optimal number is 3, so we choose to include 3-day historical prices in the input features for both NASDAQ and S&P 500 datasets. We again experimented on different training set sizes ranging from 1 year to 5 years. To form a fair comparison between the generative machine learning model, represented by SVM, and the discriminative machine learning model, represented by LR, we include daily stock prices of three major companies: Apple, Google, and Tesla, as input features in each dataset, in addition to historical prices of the two financial indices.

iii. Logistic Regression

We choose the logistic regression (LR) model to represent the category of discriminative machine learning models that we have learned in class. In theory, LR is a good model in a

binary classification problem because based on our training dataset, we can directly estimate the posterior probability using maximum likelihood estimation (MLE).
Here when we implement the LR model, we choose to use the Logistic Regression function in the Python sickit-learn library. In the first part, similar to the SVM model, we built a model solely based on the historical prices of the financial index as input features. We tested which one could achieve the best accuracy given different training set sizes, ranging from 1 year to 5 years, and different numbers of historical daily prices, ranging from 1 day to 5 days. In the table, we record the optimal number of daily historical prices with highest valid accuracy for each size of training set.

NASDAQ				S&P 500				
Training Set Size	**n**	**Valid Acc 90 days**		**Training Set Size**	**C**	**gamma**	**n**	**Valid Acc 90 days**
1 year	1	0.566666667		1 year	1.00E+03	1.00E-07	1	0.577777778
	2	0.6				1.00E-07	2	0.533333333
	3	0.655555556				1.00E-07	3	0.588888889
	4	0.522222222				1.00E-07	4	0.533333333
	5	0.555555556				1.00E-07	5	0.511111111
2 years	1	0.566666667		2 years		1.00E-05	1	0.577777778
	2	0.611111111				1.00E-05	2	0.533333333
	3	0.622222222				1.00E-05	3	0.6
	4	0.555555556				1.00E-05	4	0.533333333
	5	0.577777778				1.00E-05	5	0.555555556
3 years	1	0.566666667		3 years		1.00E-05	1	0.577777778
	2	0.622222222				1.00E-05	2	0.555555556
	3	0.655555556				1.00E-05	3	0.6
	4	0.555555556				1.00E-05	4	0.555555556
	5	0.6				1.00E-05	5	0.522222222
4 years	1	0.566666667		4 years		1.00E-05	1	0.577777778
	2	0.622222222				1.00E-05	2	0.566666667
	3	0.655555556				1.00E-05	3	0.588888889
	4	0.611111111				1.00E-05	4	0.555555556
	5	0.566666667				1.00E-05	5	0.522222222
5 years	1	0.566666667		5 years		1.00E-05	1	0.577777778
	2	0.622222222				1.00E-05	2	0.566666667
	3	0.633333333				1.00E-05	3	0.588888889
	4	0.622222222				1.00E-05	4	0.555555556
	5	0.611111111				1.00E-05	5	0.544444444

In our second part, we want to include daily stock prices as our additional input features. Same as the SVM model, the optimal number of daily stock prices is 3 for both NASDAQ and S&P 500 datasets, so we choose to use 3-day historical prices here. We again tested the impact on model accuracy of different training set sizes, ranging from 1 year to 5 years.

iv. Results and Interpretation

In the first part of experiments, when we only included the historical financial index as input in our models, we noticed that the two machine learning models, SVM and LR, generally had a better performance than the Time Series model. Moreover, SVM model achieves highest valid prediction accuracy amongst all 3 models, so we posit that generative machine learning models are better at predicting directional movement of the financial market. We continue to validate this assumption in the second part of experiments.

NASDAQ					
Training Set Size	**Test Acc 90 days**				
	Time Series	**SVM**		**LR**	
		n	**Best Acc**	**n**	**Best Acc**
1 year	0.503	4	0.659	3	0.656
2 years	0.492	4	0.637	3	0.622
3 years	0.487	3	0.615	3	0.656
4 years	0.493	3	0.573	3	0.656
5 years	0.506	3	0.703	3	0.633

S&P 500					
Training Set Size	**Test Acc 90 days**				
	Time Series	**SVM**		**LR**	
		n	**Best Acc**	**n**	**Best Acc**
1 year	0.507	3	0.626	3	0.589
2 years	0.504	4	0.571	3	0.600
3 years	0.490	3	0.571	3	0.600
4 years	0.498	1	0.551	3	0.589
5 years	0.493	3	0.626	3	0.589

Fig 1: 90 day prediction accuracy of ARIMA(3,1,2), SVM, and LR on NASDAQ and S&P 500 datasets

For the ARIMA (3,1,2) model, the mean accuracy prediction is about the same at 0.5 for different years of training data. Even though the p-value around 0.05 from the ANOVA test suggests borderline statistical significance, it might come from the large sample size, and fluctuations of accuracy at this low does not make much difference practically. Thus we can safely say that increasing the training size does not significantly lift the testing accuracy. One good thing is that the testing accuracy roughly follows a normal distribution, without many skewed datasets.

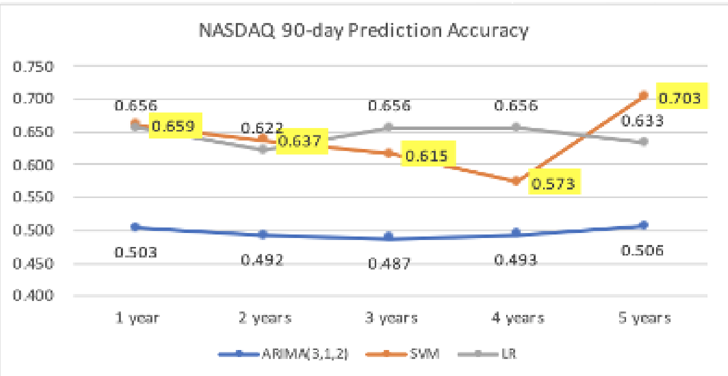


Fig 2: line plot of NASDAQ 90-day prediction accuracy of ARIMA, SVM and LR

On the other hand, both the SVM and LR model are able to achieve average accuracy of around 60%. Meanwhile, we observed that the two models performed better on the NASDAQ dataset than the S&P 500 dataset. For SVM, we have witnessed best valid accuracy performance when the training set is of size of 5 years on both NASDAQ and S&P 500 datasets, indicating that larger training dataset could contribute to better prediction accuracy performance. Interestingly, for LR, while training accuracy initially improved when training set size increased from 1 year to 2 years, a larger training dataset size also did not necessarily lead to higher valid accuracy. Specifically, we observe a decrease in valid accuracy for LR after training set size reaches 2 years. This could be because that since fluctuations in the financial market are noisy and are related to many factors, current regressional relationship between historical daily prices may not correlate much to that from many years ago.

In the second part of our project, we include historical prices of three major stocks, Apple, Google, and Tesla, as additional input features to further test on prediction accuracy of SVM and LR models. We choose to include the optimal number of 3-day historical prices, that we obtained in the first part of experiments, in the input features for both SVM and LR models on both NASDAQ and S&P 500 datasets. We noticed that for both NASDAQ and S&P 500 datasets, SVM model gives best valid prediction performance with a training set size of 5 years, again indicating that larger training dataset could contribute to better prediction accuracy performance. Interestingly, for LR, we again observe a decreasing trend in valid accuracy when training set size increased, again proving the point that current regressional relationship between historical prices can't be determined by historical regressional relationships from many years ago. In general, among both models, SVM is again the one with the best prediction performance, validating our assumption before that generative machine learning models, like SVM, are better than discriminative machine learning models, like LR, at predicting directional movement of the financial market.

NASDAQ				S&P 500		
Training Set Size	Test Acc 90 days			Training Set Size	Test Acc 90 days	
	SVM	LR			SVM	LR
1 year	0.615	0.633		1 year	0.615	0.611
2 years	0.560	0.622		2 years	0.549	0.556
3 years	0.571	0.622		3 years	0.582	0.600
4 years	0.593	0.611		4 years	0.604	0.611
5 years	0.640	0.589		5 years	0.652	0.600

Fig 3: 90-day prediction accuracy of SVM and LR with historical price

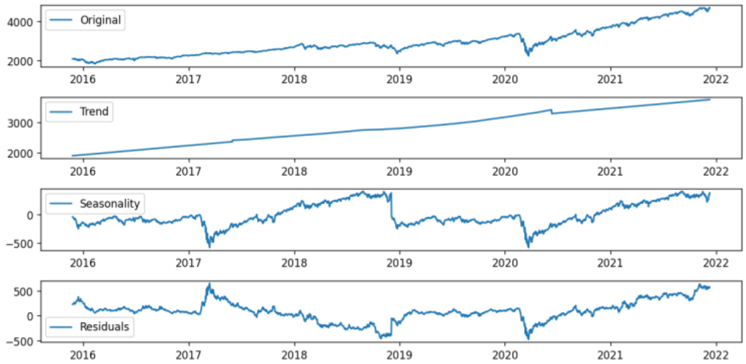
V. Conclusion

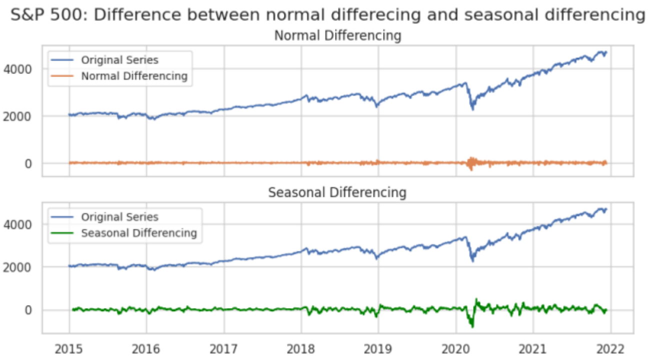
In our project, we explored three different models' performances to predict directional movement of financial indices prices. We first established a fair comparison between the econometric time series model and two machine learning models, proving that machine learning models are generally better at the prediction task. Then, we went on to examine the ability of the two different types of machine learning models- generative and discriminative- to predict financial indices price movements. We discovered that generative machine learning models, represented by Support Vector Machine, achieve higher prediction accuracy than discriminative machine learning models, represented by Logistic Regression. Through extensive controlled experiments on two different datasets, we succeeded in establishing the outcome of this project as valid and trustworthy.

VI. Further Exploration

Time Series: SARIMAX

The problem with the ARIMA is that it does not support seasonality. For this part, we want to take into account the seasonal effects. From the plots of S&P 500 data decomposed below, we can see that seasonality does follow some patterns after we decompose the financial index prices. The same applies to the NASDAQ dataset.





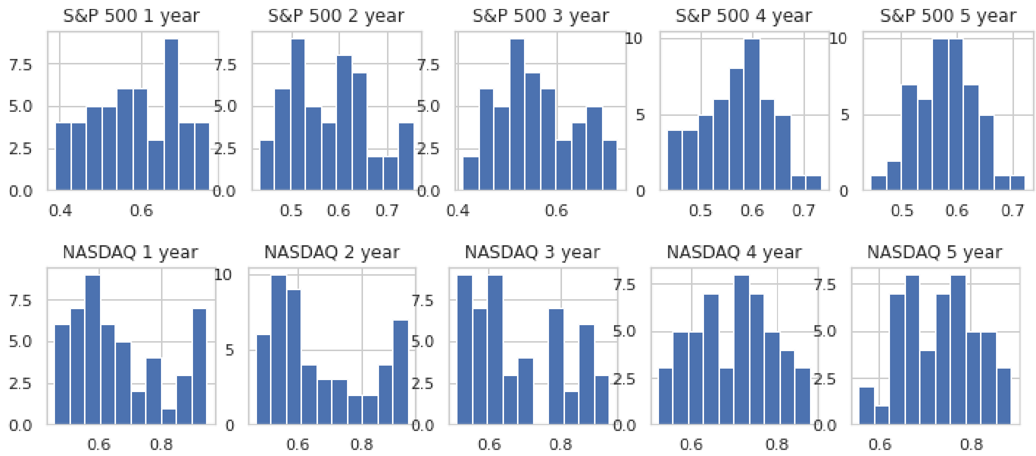
And from the left plots, it is clear that seasonal differencing can capture more variances in the data more accurately.

So we want to see if the SARIMAX model with exogenous variable could improve the testing accuracy.
As usual, we use auto_arima to find the optimal combination of parameters. It turns out the best model is SARIMAX(3,0,0)*(2,1,0,12).

The following is the table for SARIMAX 90 day test accuracy.

Test Acc 90 Days	**SARIMAX(3, 0, 0)x(2, 1, 0, 12)**	
Training set size	**NASDAQ**	**S&P 500**
1 year	0.6888888888888889	0.5866666666666667
2 year	0.6997777777777778	0.574
3 year	0.7033333333333335	0.5964444444444444
4 year	0.7451111111111112	0.574888888888889
5 year	0.7317777777777779	0.5806666666666668
P-value of ANOVA test	0.11483057981122863	0.6977378006196855

SARIMAX can achieve a significantly higher prediction accuracy, which suggests that seasonality plays a big part in financial index changes. However, the prediction accuracy among various year ranges is still not so different. Moreover, SARIMAX works better on NASDAQ than S&P 500.


S&P 500 1 year
S&P 500 2 year
S&P 500 3 year
S&P 500 4 year
S&P 500 5 year
NASDAQ 1 year
NASDAQ 2 year
NASDAQ 3 year
NASDAQ 4 year
NASDAQ 5 year