Final Project Writeup

Financial Commodity Time Series Price Prediction

Kruti Mody, Mohit Shah, Amol Soley, Shachi Sonar, Venkat Srinidhi Vaddy

Overview:
In financial and academic areas, stock market and financial commodity price analysis and prediction is extremely crucial. While the fundamental analysis focuses on the business model of a company and future developments, technical analysis focuses only on the price movement of the stock and relevant variables to predict price.

Time series forecasting is a technique involving collection of data at regular periods and analyzing both the data and the time sequence to predict future events and results. With a two fold purpose of studying the old historical data and drawing insights as well as forecasting future values based on historical information, time series analysis helps in understanding seasonality, patterns, trends, cyclicity and anomalies or irregularities.

The financial market is influenced by the popularity, trends in market and economic environment, exchange rates, policies around financial markets, gold and silver prices as well as other factors including restrictions, legalities, supply and demand rates and circulation of the commodities. Besides the market related variables - opening and closing price, trade volume, short and long term averages, external determining variables have to be taken into account while establishing a price prediction mechanism for potential investors to make smart data-driven decisions.

Motivation:
Stock markets are high risk investments that are influenced by a long list of factors and trends along with seasonal variance. This research will be helpful to the investors in the cryptomarket/stock market by providing them with a model that would be taking into account these variances and providing them with historical data, future predictions and trends that would lead them to make a more informed decision. They would be able to weed the outliers and fluctuations and get an overview of the actual trend and pattern that the commodity is following. If we are successful, we would be minimizing the risks involved with trading and investing in the stock market. Rather than making random decisions or ill informed decisions, potential investors would be more knowledgeable while investing in the stock market.

Thus, financial reporters, institutions, and public investors would be provided with comprehensive information regarding the financial risk associated with investing in the stock market helping them to get a better grasp of how their commodities are doing and what the future repercussions would be.

The idea for this project comes from a shared interest of our group in the finance sector, and a curiosity into the volatile nature of the finance industry. Collectively we were keen on identifying the most predictable and stable way of predicting commodities, which could be used for directing personal investment options as well.

While time-series forecasting algorithms being applied on the stock market and cryptocurrency is not a novel problem, but what we are aiming from this research is:

- **Identifying the best algorithm** – with multiple algorithms in the space, it is important to identify which ones have higher reliability and why
- **Identifying predictable commodities** - which are typically more likely to be forecasted with accuracy, and showcase stability as investment options

Problem statement:
We had 3 main goals that we targeted to achieve through our project:

a. Understanding the impact of macroeconomic factors like Inflation, GDP and unemployment rate on the commodity's price.
b. How reliably we can make predictions to make personal investment decisions?
c. Compare and contrast the performance of various models such as LSTM, ARIMA, SARIMA and fbProphet in making predictions.

Related work: Literature Review and Previous work
Techniques like simple moving average and exponential moving average are widely used in current models. Current models for bitcoin price prediction uses Linear regression and LDA with an accuracy of 60% and 65% respectively (Chen et al., 2020). Ordinary regression models will use time indices as x/independent variable. Linear Regression (LR) is not the best method to use when making predictions on time series data as LR assumes that residues are not correlated but this is not the case with time series data.

Previous work suggests that ARIMA-LSTM show best performance in stock index prediction. ARIMA primarily addresses linearity of model whereas LSTM addresses non-linearity of model using recursive neural network. Financial anomalies and noise are common, political, economical and other all special trends cannot be covered by the model. (Xiao & Su, 2022). Several works suggested that ARIMA does not take seasonality into consideration. So we did a comparative study of ARIMA, SARIMA, LSTM and Prophet to find an effective hybrid model.

Dataset:

We gathered our data primarily from 2 sources - *Yahoo Finance* and *US bureau of labor statistics.* The collected data was taken within a time span of 2006-2022. The data collected from Yahoo Finance comparized of details like Commodity type, High, Low and adjusted Close for 29 stocks, 25 cryptocurrencies and 100 ETFs. We completed this step by making use of the get_data_yahoo() function in our python script. The data comparing Inflation rate, Monthly Employment Rate and Quarterly GDP was extracted from US bureau of labor statistics in the same time frame of 2006-2022.

<u>Dataset 1: Stocks, ETFs and Cryptocurrencies</u>

- Source: Yahoo finance
- Collection approach: Scraped using python script
- Details:
  - 29 Stocks listed on the NYSE (2006-2022)

- 25 Cryptocurrencies (2014-2022)
- 100 ETFs (2006-2022)

- Features: Date, Adjusted Close, High, Low, Volume, Commodity, Type

Dataset 2: Macroeconomic factors

- Source: US bureau of labor statistics
- Collection approach: Pulled from website data tools
- Details:
  - Monthly CPI-U (2006-2022) - Inflation
  - Monthly Unemployment Rate (2006-2022)
  - Quarterly GDP (2006-2022)
- Features: Year, Period, Value, Series ID

Approach:

Data Transformation and Metrics

After having retrieved the data, the next important step was to clean it to fill null values. The null values exist in the data set for the following reasons:

- Stock ceased to exist or the company expired
- Certain commodity types were blank, because the data was unavailable.

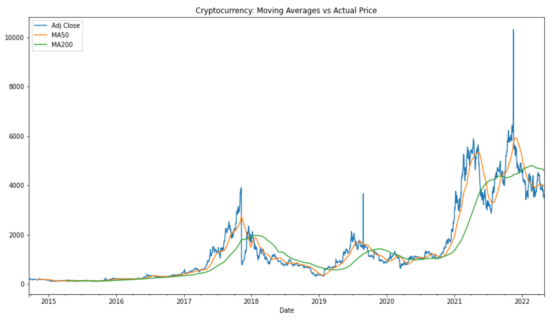
To fill data for these entries, we filtered out commodities based on the date range which we had majority data points for. There were some instances where commodity types were blank, so we had to manually identify which commodities were these and fill them manually.

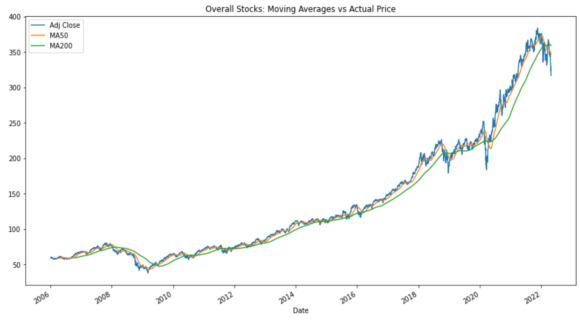
We also had disparate data sources, with GDP being quarterly, CPI & unemployment monthly and commodities daily, to join data we had to align dates. We used ffil() methods to the approach of assuming GDP same for the entire quarter and similarly CPI/unemployment index the same for the whole month to join with the daily data for commodity prices. For univariate models like ARIMA, SARIMA, LSTM, Prophet, the “closing price” was the single metric which was important for our use. For multivariate models, we used closing price, GDP, unemployment rate and CPI for training the model.

The primary metric for evaluation was RMSE (Root Mean Squared Error). It is an absolute error measure that squares the deviations to keep the positive and negative deviations from canceling one another out. This measure also tends to exaggerate large errors, which can help when comparing methods. Based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately for time series.

Commodity trends

We analyzed overall trends for each commodity type - specifically for stocks and cryptocurrency.



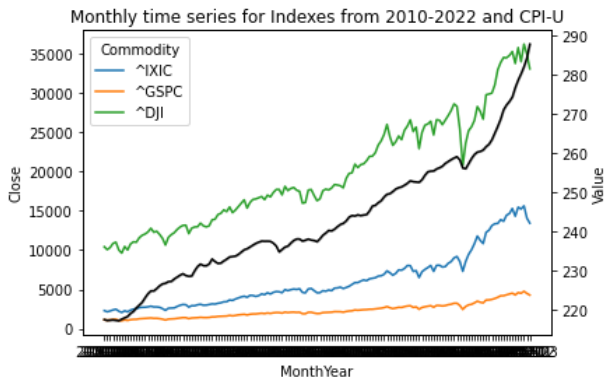


There are 3 lines, cryptocurrency price (average), moving average 50 days and 200 days, and we can observe clearly that stocks are way harder to predict v/s stocks. The curves are more gentle on the chart to the right. This set the initial expectation for our analysis that stocks will be more likely to be predictable and cryptocurrencies will be harder to predict given their volatility.

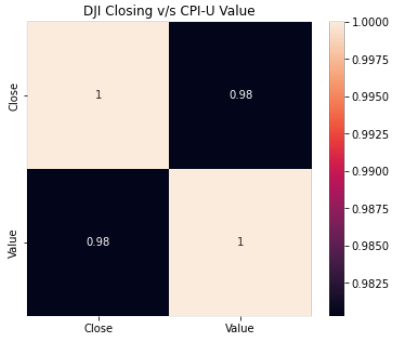
Macroeconomics

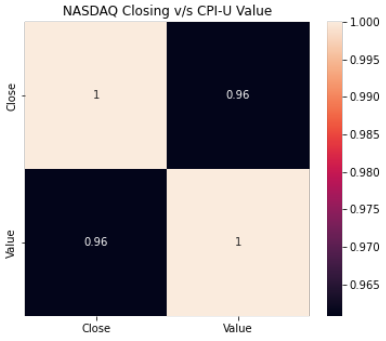
Macro economic factors such as unemployment rates, GDP, and inflation rates also have an impact on the trends of the commodities market, which can be useful indicators for prediction of prices overall. Below we compare how the top indexes in the US stock market, such as NASDAQ, DJI and S&P 500 behave in relation to these factors:

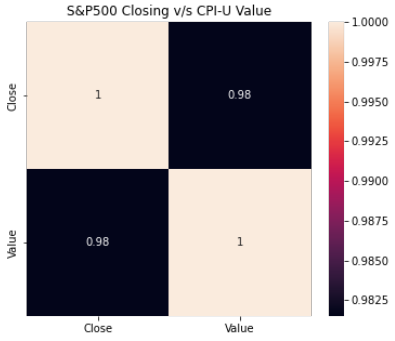
a. <u>CPI-U</u>



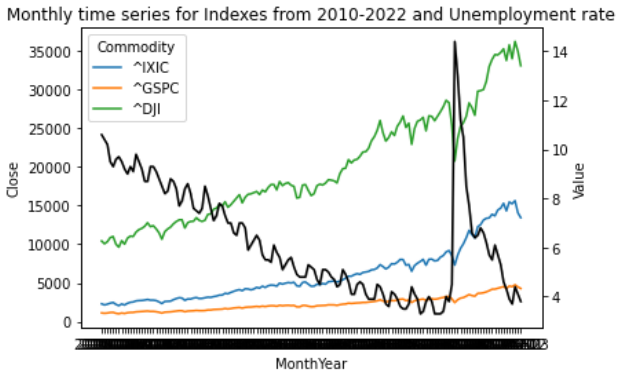
We observe very high correlation between the variables based on the factors listed above. Indicating that they can be potentially useful for prediction of the stock market.



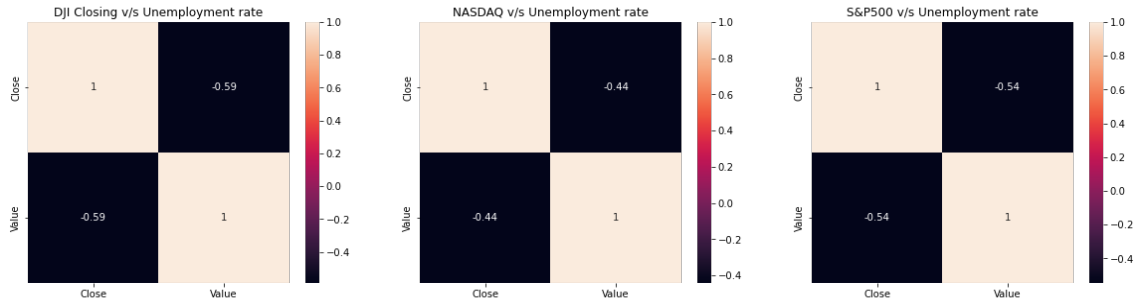




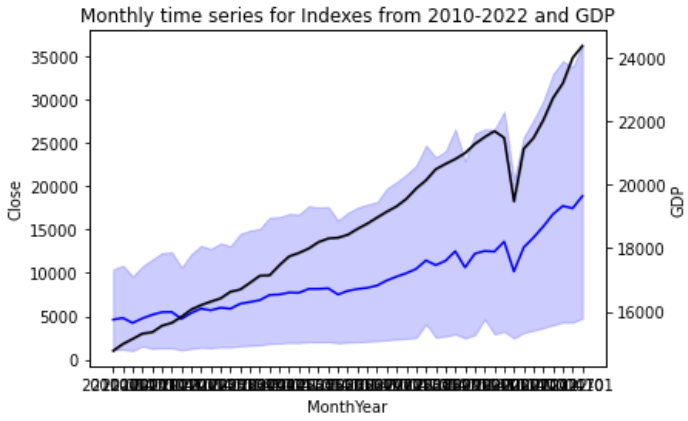
b. Unemployment Rate



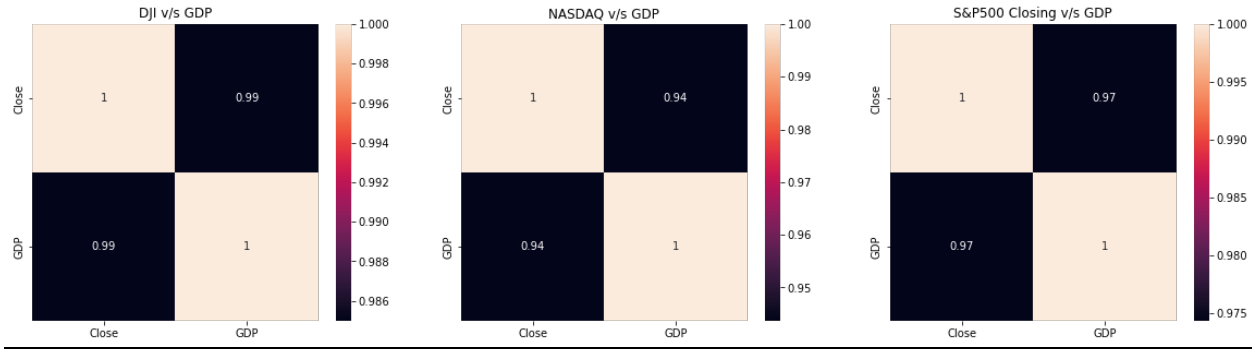
In comparison to the CPI-U, we notice that the unemployment rate is not so strongly correlated. We do see a moderate negative correlation, which makes sense based on our findings. With increase in unemployment, it is understandable that the stock market will have lesser investors and thus, less growth. That, however, is not the strongest of correlations. In fact we do observe a huge spike in the unemployment rate in 2020, which is due to covid (we do see a crash in the stocks in the same period as well)



c. GDP



For GDP, it is understandable that as GDP will rise, the market overall for the country improves. Which is also evident based on the correlation between the two.



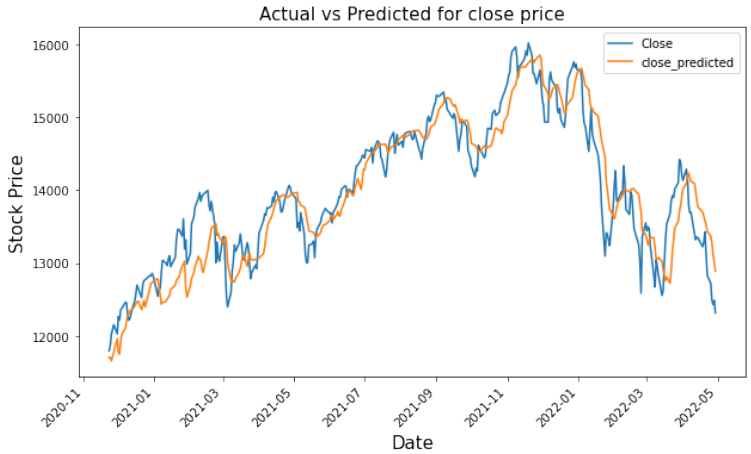
Experiments:

LSTM

The Long Short-Term Memory (LSTM) model is a recurrent artificial neural network which consists of a feedback mechanism thus making it different from the traditional feed forward neural networks. As it has the ability to process long sequences of data rather than just single data points, it is widely used in areas of speech recognition, video processing, as well as stock market data. This is due to the architecture of the network which consists of a cell and multiple gates which aid it in 'remembering' data. It can be thought of as a step up from the regular recurrent neural networks which keep track of long-term sequences making them to be computationally heavy as LSTM consists of a 'forget' gate which assists in forgetting information that is no longer needed.

There are three types of gate:

1. Forget Gate
    a. It decides which information to throw away.
2. Memory Gate
    a. Which values to update into the memory.
3. Output Gate
    a. What should be the output based on the input as well as the already present memory.



For experimentation, we trained our LSTM model on the NASDAQ index data with the following architecture:

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_12 (LSTM)              (None, 50, 50)            10600

 dropout_6 (Dropout)         (None, 50, 50)            0

 lstm_13 (LSTM)              (None, 50)                20200

 dense_6 (Dense)             (None, 2)                 102

=================================================================
Total params: 30,902
Trainable params: 30,902
Non-trainable params: 0
_________________________________________________________________
```

Using the closing data values, we created sequences of the data and then input it into our model. We achieved a training accuracy of 97.95% and validation accuracy of 98.36%

Forecasting with ARIMA

ARIMA (Auto Regressive Integrated Moving Average) is a time series prediction model which makes predictions based on the past values. It is useful for predicting short-term market movements and works best on a linear time-series data where the current value is influenced by and predicted based on the previous value. It is a combination of linear regression between past fitted values with present fitted values and linear regression between present and past values of residuals (moving average).

It calculates the values as follows -

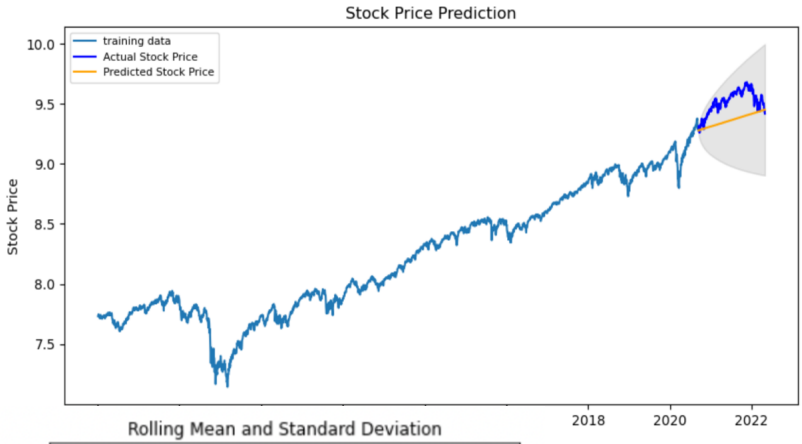
$$mt = \beta 0 + \beta 1 mt\text{-}1 + \beta 2 mt\text{-}2 + \beta 3 mt\text{-}3 + \ldots + \beta p mt\text{-}p$$

$$mt = \beta 0 + \beta 1 et\text{-}1 + \beta 2 et\text{-}2 + \beta 3 et\text{-}3 + \ldots + \beta q et\text{-}q$$

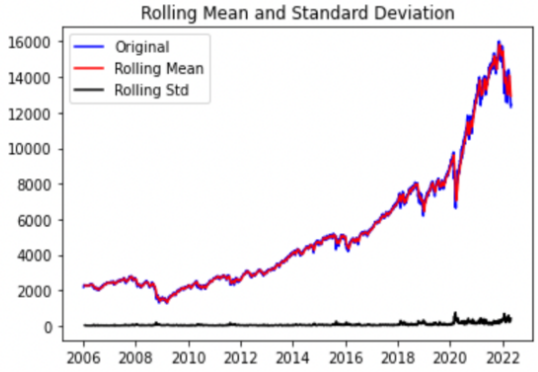
$$e = \text{error term}$$

Where, p is the number of lag observations, d is the number of times the values have been differenced and q is the size of the moving average window.

The ARIMA model is fit to the stationary training data and predicts on testing data. It does not take into account seasonality at all.



The RMSE value of this model is 0.16. With MAPE of around 1.5%, the model is 98.5% accurate in predicting the new observations (just based on previous values and not considering external factors or seasonality, which in reality greatly influence the market and price)




```
Results of dickey fuller test
Test Statistics                  1.016777
p-value                          0.994440
No. of lags used                27.000000
Number of observations used    4082.000000
critical value (1%)              -3.431953
critical value (5%)              -2.862248
critical value (10%)             -2.567147
```

Test for Stationarity and Seasonality

As the data for ARIMA has to be stationary, we ran an Augmented Dickey Fuller Test (ADF) to understand if the data is stationary. The null hypothesis states that the series is non-stationary and as predicted the p value for a level of significance of 0.05 was greater. Thus, we transformed our data to be stationary in order for it to be considered as valid input for ARIMA.

Forecasting with SARIMA

While ARIMA model can work with trends, the problem with ARIMA is that it does not support seasonal data - that is time series with a repeating cycle. Therefore, we use Seasonal ARIMA. The Seasonal Autoregressive Integrated Moving Average model is a method of time series forecasting with univariate data containing trends and seasonality.

SARIMA adds additional components of seasonal autoregression, moving average and additional parameter for period of seasonality.

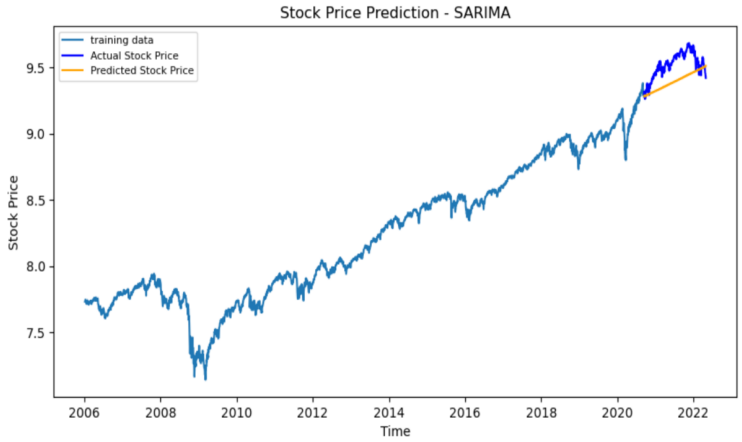
Seasonal autoregression is a regression of the variable against itself - for historical data and for seasonality. The lagged values of the target variable are taken as input variables to forecast values for the future. Similarly, offset seasonal values are taken as input variables to forecast seasonality.

$$mt = \beta 0 + \beta 1 mt\text{-}1 + \beta 2 mt\text{-}2 + \beta 3 mt\text{-}3 + \ldots + \beta p mt\text{-}p$$

Moving average component uses past forecast errors than the past values in regression models to forecast future values. Similarly for seasonality.

$$mt = \beta 0 + \beta 1et\text{-}1 + \beta 2et\text{-}2 + \beta 3et\text{-}3 + \ldots + \beta qet\text{-}q$$

The hyperparameters for both trend and seasonality include p (trend autoregression order), d (trend difference order), q (trend moving average order), P (seasonal autoregressive order), D (seasonal difference order), Q (seasonal moving average order) and m (shows the seasonal cycle). If m is 12, It shows annual seasonal cycle. P is how many previous values or offset observations need to be taken for future value forecasting. Q is how many previous forecasting error terms need to be taken in the equation.



One instance of the model performance: SARIMA model was trained on IJIX index. The RMSE value of this model is **0.145**. With Mean absolute percentage error as 1.3%, the model is 98.7% accurate in predicting the new observations. While it accounts for seasonality, **hypertuning** parameters is very crucial for SARIMA model to improve performance.

Multivariate Vector Autoregression

In multivariate time series, we have more than one time dependent variable such as GDP, stock price, inflation index and unemployment index. Each variable is dependent on its own past values as well as is influenced by past values of the other variables. This dependency is used to forecast future values.

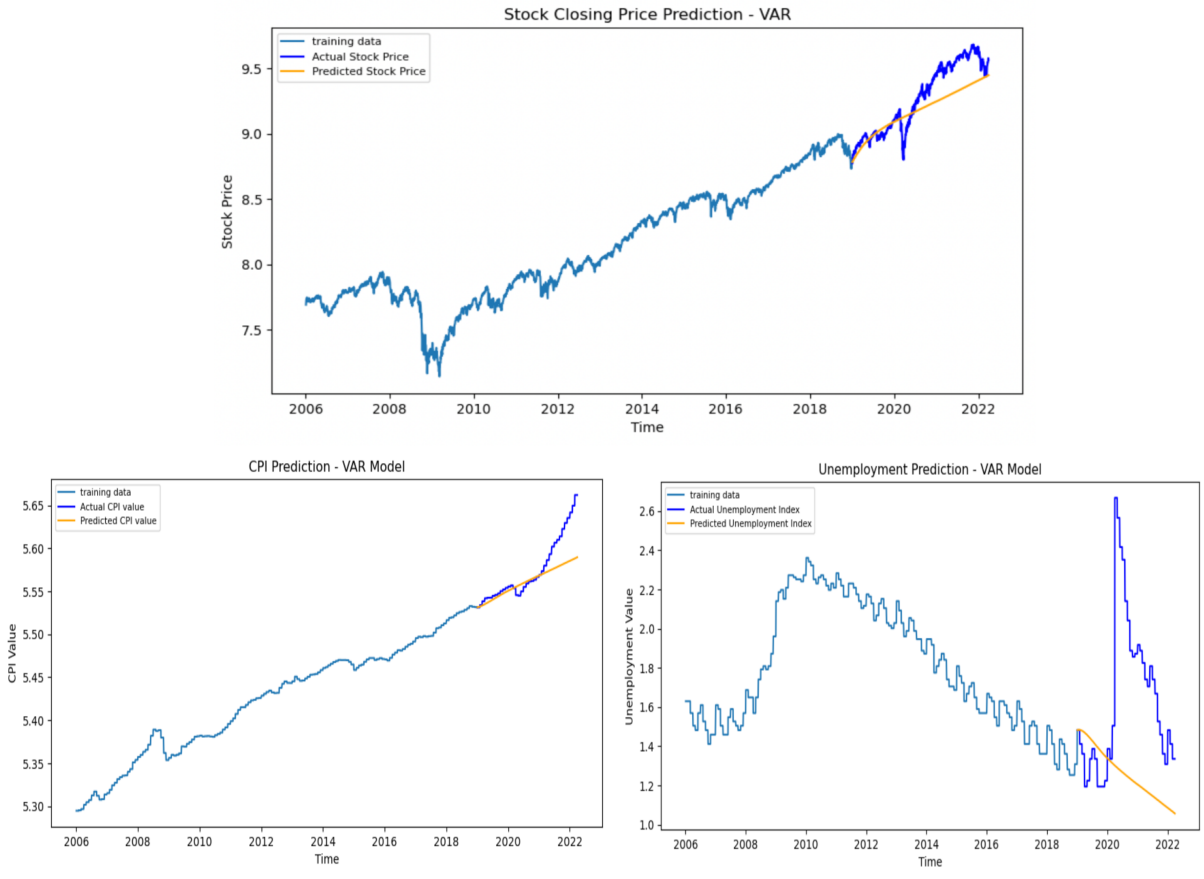
We use the Vector Auto Regression method which uses a linear function of forecasting using past values of all 4 variables to predict all. It focuses on bidirectional dependency - if x influences y, then also uses y influence on x.

It also includes Granger Causality test which states that there is no cause and effect relationship between the variables and generates p value. If p value < 0.05, row variable influences column variable. For instance, highlighted in green, CPI, unemployment and GDP all influence stock closing price.

	Close	CPI	Unemp	GDP
Close	1.0	0.02	0.008	0.057
CPI	0.0	1.0	0.87	0.3
Unemp	0.0	0.16	1.0	0.11
GDP	0.0	0.04	0.11	1.0

Fig. Result of Granger Causality Test

The VAR model was trained on GDP, CPI, Unemployment Index and Closing Price of stock market index. Using train-test split, the rmse value of the model is generated to see model performance in forecasting values based on historical data and influence of other variables.



Prediction Results:

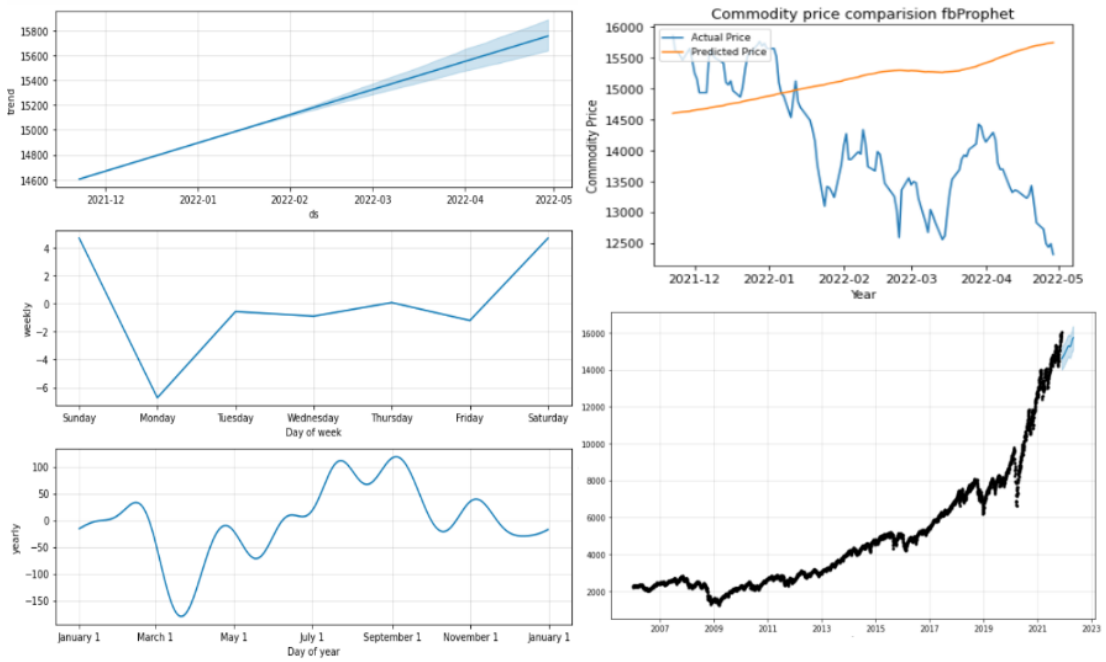
Series	RMSE
Stock (Closing)	0.15615431
CPI	0.02466444
Unemployment	0.57566916
GDP	0.04057319

The RMSE for closing price has increased from 0.14 in the SARIMA model to 0.15 here due to the influence of external factors and variables.
The model performs well in predicting the future values of the macroeconomic factors as well based on the influence and dependency of stock prices and other macroeconomic factors on each other.
But unpredictable world events such as Covid outbreak reduces model performance. And can't be accounted for through the model or be trained into the model. Fig 3 in VAR predictions show accurate predictions in the unemployment index in 2020 based on historical values but when COVID hit in 2020, there was a sudden spike in the unemployment index which cannot be predicted. This is the biggest limitation of the time series models.

However, the multivariate VAR model gives a holistic true picture of the time series forecasting which heavily includes interdependencies.

Forecasting with Prophet Model:
Prophet (earlier FbProphet) is a time series forecasting algorithm released by the data science team by facebook and used by facebook to produce reliable forecasts for planning and goal setting. The algorithm is based on an additive model (trend + seasonality + holiday + error) where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

Here we see general, weekly and yearly trends. But it has poor accuracy for our chosen dataset.



Results, Discussion and Findings

Here, we have Model performance measures for our core models. We ran all our models through each stock, ETF and cryptocurrency and picked the “best” and “worst” performing stocks in terms of prediction, clearly LSTM works best and prophet is least effective. The graph also shows changes in commodity prices owing to unemployment, covid-19, inflation and GDP.

RMSE for commodities based on different algorithms					
Type	**Commodity / Model**	**ARIMA**	**SARIMA**	**LSTM**	**PROPHET**
Stock	**Apple – AAPL**	**0.34**	**0.07**	**0.004**	**52.64**
	Google – GOOGL	**0.34**	**0.33**	**0.005**	**857.32**
	Cisco – CSCO	**0.17**	**0.16**	**0.014**	**15.81**

Cryptocurrency	**Bitcoin – BTC USD**	**0.23**	**0.22**	**0.05**	**21275.58**
	Doge – DOGE USD	**0.82**	**0.9**	**0.005**	**0.13**
ETF	**SPDR S&P 500**	**0.15**	**0.14**	**0.01**	**125.05**
	Vanguard Midcap	**0.18**	**0.18**	**0.006**	**69.47**

Recommendation:

There are two types of time series forecasting:

a) Univariate - just the time component to forecast the target variable
b) Multivariate - time as well as other independent factors to forecast target variable.

We performed univariate forecasting and looked at the metrics such as root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and believed that by incorporating external components like inflation, unemployment, GDP, etc one could build a multivariate model to improve performance. We did a quick analysis using multivariate vector regression and adding regressors to the prophet model and used the metrics to strengthen the above hypothesis and recommendation.

Limitations, Future Work and Conclusion

This work comes along with certain limitations, which do need to be addressed:

- Unpredictability: Volatility is high in the stock market, and forecasting algorithms may not be the best to predict unforeseen market crashes, like during covid, 2008 housing crisis, etc.
- Long Term Forecasts: Forecasts from our results may be best suited to make short term investment decisions, which are based on more regular patterns for pricing
- Expertise and Hyperparameter Tuning: Collaboration with financial experts who have researched the field would be more valuable, as there can be addition of more important variables and parameters to our algorithms
- Ethical challenges: Adding company specific data can improve predictions, but financial data must be used with care as it can have confidentiality issues. Also ethical issues exist by providing financial advice through such tools, which can potentially still harm people if it has prediction errors

Conclusion: We have performed an end to end analysis of performing a time series prediction by understanding the literature around it, gathering, transforming and cleaning relevant data. After running them through multiple algorithms, our recommendations are that univariate algorithms

may be good for a stable market where unpredictable events are very rare, primarily because of how they perform with accuracy. However, multivariate algorithms may be more suited to predict stock prices better as they incorporate multiple economic factors which influence commodities.

References:

1. Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. Journal of Computational and Applied Mathematics, 365, 112395. https://doi.org/10.1016/j.cam.2019.112395

2. Dutta, A., Kumar, S., & Basu, M. (2019). A Gated Recurrent Unit Approach to Bitcoin Price Prediction. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3514069

3. Iqbal, M., Iqbal, M., Jaskani, F., Iqbal, K., & Hassan, A. (2021). Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques. *EAI Endorsed Transactions on Creative Technologies*, *8*(28), 170286. https://doi.org/10.4108/eai.7-7-2021.170286

4. M. Ahmed et al. (2017) "Anomaly Detection on Big Data in Financial Markets," 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

5. T. (2018, August 2). Everything you can do with a time series. Kaggle. https://www.kaggle.com/code/thebrownviking20/everything-you-can-do-with-a-time-series/data

6. Xiao, D., & Su, J. (2022). Research on Stock Price Time Series Prediction Based on Deep Learning and Autoregressive Integrated Moving Average. Scientific Programming, 2022, 1–12. https://doi.org/10.1155/2022/4758698