AI60003: ARTIFICIAL INTELLIGENCE FOR ECONOMICS

Term Project Report

Comparative Study of stock price forecasting using ML models on global stocks

Group members:

1. Amit Kumar (20CS30003)
2. Anand Manujkumar Parikh (20CS10007)
3. Atishay Jain(20CS30008)
4. Likhith Reddy (20CS10037)
5. Shivansh Shukla (20CS10057)
6. Vaibhav Maheshwari(20HS20055)

Introduction

Stock markets serve as electronic marketplaces where individual and institutional investors engage in buying and selling shares. The dynamic determination of security prices relies on the principles of supply and demand, reflecting the levels at which market participants are willing to transact. Predicting stock market movements involves anticipating the future value of company stocks or financial instruments traded on exchanges. Successful predictions can lead to significant profits. While traditional approaches heavily lean on statistical and econometric models, these methods often fall short in addressing the dynamic and complex nature of the stock market. The integration of Artificial Intelligence (AI) and Machine Learning (ML) tools has emerged as a promising solution for overcoming these challenges in stock price prediction. This study delves into time series analysis, offering foundational insights into this evolving field and paving the way for further research in the stock market sector.

Problem Statement

This project aims to review various methods of stock price prediction, focusing on a subset of algorithms from both statistical and machine-learning based approaches. A comparative analysis of these algorithms' performance is conducted to evaluate their efficacy in forecasting stock prices.

Literature Review

The first paper, titled "Comparative Study of Machine Learning Models for Stock Price Prediction," aims to apply machine learning techniques to historical stock prices for forecasting future prices. The study employs recursive approaches suitable for time series data, specifically utilising a linear Kalman filter and various long short-term memory (LSTM) architectures. Historical stock prices spanning a 10-year range (1/1/2011 – 1/1/2021) are utilised for model training and evaluation. The performance of the models is quantified by computing the error between predicted and historical values for each stock. Notably, the study finds that a simple linear Kalman filter performs well for low-volatility stocks, such as Microsoft ('MSFT'), while complex LSTM algorithms significantly outperform the Kalman filter for high-volatility stocks like Tesla ('TSLA'). The results indicate the potential for classifying and training LSTMs for different stock types, offering a method to automate portfolio generation based on stock characteristics.

The second paper, titled "Implementation of Kalman Filter with Python Language," focuses on the implementation of a Kalman Filter using the Python language, specifically leveraging the Numpy package. The Kalman Filtering process is outlined in two steps: Prediction and Update. The paper introduces functions coded with matrix input and output for each step, offering a practical and functional implementation. An illustrative example of a Kalman Filter application is provided, demonstrating its usage for the localization of mobile devices in wireless networks. The paper contextualises the Kalman filter within the broader landscape of mathematical tools for stochastic estimation, emphasising its significance in navigation applications. The recursive and optimal nature of the Kalman filter, minimising error covariance under specific conditions, contributes to its widespread use in the field of autonomous or assisted navigation.

Data Sources

The stock market data is collected using Yahoo Finance API. We collected historical stock price data of Apple Inc, MicrosoftInc, Tesla Inc. for about 6 years between the time period April 02, 2017 to November 27, 2023.

Apple Inc

Date	Open	High	Low	Close	Volume
2017-04-03 00:00:00-04:00	33.580837830663064	33.6766401648162	33.426614221244776	33.57849884033203	79942800
2017-04-04 00:00:00-04:00	33.473344899162036	33.85656490058694	33.45465080573476	33.82852554321289	79565600
2017-04-05 00:00:00-04:00	33.70000798986088	33.989761107829246	33.60420209212126	33.65327453613281	110871600
2017-04-06 00:00:00-04:00	33.716359201303895	33.770106054142865	33.52007657898831	33.569149017333984	84596000
2017-04-07 00:00:00-04:00	33.58550862062783	33.69065978485032	33.4780220316125	33.49437713623047	66688800

Microsoft Inc

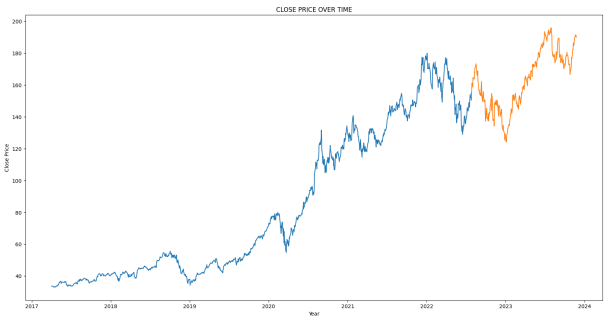
Date	Open	High	Low	Close	Volume
2017-04-03 00:00:00-04:00	60.452164345266716	60.57158505386758	59.88264530396739	60.21333694458008	20400900
2017-04-04 00:00:00-04:00	60.06635819346271	60.452162761316266	59.96531313879763	60.37868118286133	12997400
2017-04-05 00:00:00-04:00	60.902266293226994	60.94819139708911	60.11228162093247	60.22250747680664	21448600
2017-04-06 00:00:00-04:00	60.25926047738949	60.6818093385992	60.1490346055784	60.37868118286133	18103500
2017-04-07 00:00:00-04:00	60.48890640197155	60.58995145554631	60.11228966358523	60.33274841308594	14108500

Tesla Inc

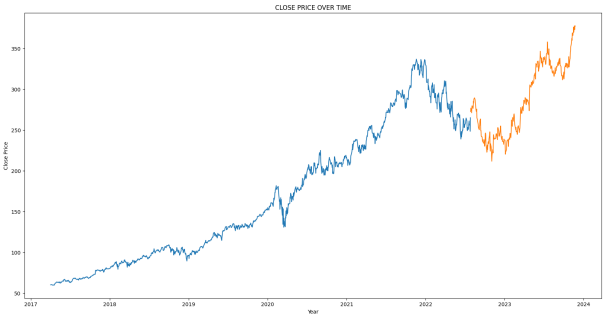
Date	Open	High	Low	Close	Volume
2017-04-03 00:00:00-04:00	19.126667022705078	19.933332443237305	18.972000122070312	19.90133285522461	208329000
2017-04-04 00:00:00-04:00	19.792667388916016	20.320667266845703	19.635332107543945	20.246667861938477	152019000
2017-04-05 00:00:00-04:00	20.13599967956543	20.325332641601562	19.613332748413086	19.666667938232422	118213500
2017-04-06 00:00:00-04:00	19.79199981689453	20.12933349609375	19.606666564941406	19.913333892822266	82809000
2017-04-07 00:00:00-04:00	19.833332061767578	20.179332733154297	19.809999465942383	20.16933250427246	68694000

The dataset contains 5 features: Open, High, Low, Close, Volume. We've used the closing price as the target variable.

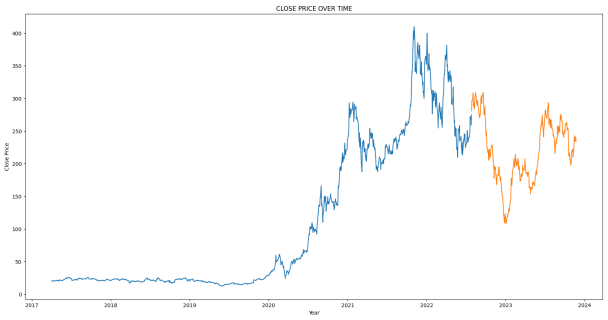
Apple Inc



Microsoft Inc



Tesla Inc



Background

LINEAR REGRESSION

Linear regression involves modelling the relationship between two variables by fitting a linear equation to observed data. In this context, one variable serves as the explanatory variable, while the other functions as the dependent variable. The linear regression algorithm is employed to construct a forecasting model, and it is widely applied due to its natural adaptability to even intricate forecasting tasks.

In the linear regression algorithm, the objective is to learn how to generate a weighted sum from input features. For a scenario involving two features, the formulation becomes:

target = weight_1 * feature_1 + weight_2 * feature_2 + bias

Throughout the training process, the regression algorithm acquires values for the parameters—weight_1, weight_2, and bias—that best align with the target. These weights are commonly referred to as regression coefficients, and the term "bias" is also known as the intercept. The intercept essentially indicates the point where the graph of this function intersects the y-axis.

MODEL

Sample Equation:

$$y_n \approx f(\boldsymbol{x}_n) = \boldsymbol{w}^\top \boldsymbol{x}_n \quad (n = 1,2,\ldots,N)$$

Loss Function:

$$L(\boldsymbol{w}) = \sum_{n=1}^{N} \ell(y_n, \boldsymbol{w}^\top \boldsymbol{x}_n)$$

$$L(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2$$

XGBOOST

XGBoost is a machine learning algorithm that relies on decision trees within an ensemble framework, utilising gradient boosting. In scenarios where prediction tasks revolve around unstructured data, such as images or text, artificial neural networks generally exhibit superior performance compared to other algorithms or frameworks. Nevertheless, for small-to-medium-sized structured or tabular data, decision tree-based algorithms are currently acknowledged as the leading choice. This underscores the notion that in cases where data is well-organised and tabulated, XGBoost, leveraging decision trees, stands out as a top-performing solution.

- Optimization form:

$$\min \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss Complexity of the trees: Regularizer

ARIMA-GARCH

At the start of the model we have decided the next stock price to be purely a function of its past values. In the case of ARIMA we first make the series stationary then we plot PACF and ACF to find AR and MA terms. One another way of finding ARIMA models is to use Information criteria such as AIC or BIC. After finding the ARIMA model we find the error term to have periods of high and low volatility for which we will be using the GARCH model to forecast.

auto_arima: It first checks for stationarity then varies p,q to minimise AIC

Procedure:

STEP-1: Find ARIMA model and predicted values to find error

STEP-2: Use the errors to create a GARCH model and find next forecasted values

STEP-3: Sum up the forecasted errors forecasted ARIMA values to find predicted values.

In our model to keep the run time low we fit the ARIMA model once a month and predict volatility every day.

MODEL EQUATIONS:

Sample Equation of ARIMA(0,1,1):

$\hat{Y}_t = \hat{Y}_{t-1} + \alpha e_{t-1}$

Sample equation of ARCH(1):

$$
\begin{aligned}
r_t &= \sigma_t e_t \\
e_t &\sim \texttt{white noise}(0,1) \\
\sigma_t &= \sqrt{\omega + \alpha_1 r_{t-1}^2}
\end{aligned}
$$

LSTM

Long Short-Term Memory (LSTM) is a specialised recurrent neural network architecture designed to address the vanishing gradient problem, commonly associated with traditional RNNs by utilising memory cells and gating mechanisms, including input, forget, and output gates, to effectively capture and retain long-term dependencies in sequential data.
LSTMs are recognized for their predictive quality in modelling complex patterns and making accurate predictions based on historical information. Their ability to learn from and adapt to extended sequences makes them particularly effective in scenarios where understanding context over time is crucial, although their performance can be influenced by factors like hyperparameter tuning and the nature of the specific task at hand.

Ensemble

Ensemble learning is a sophisticated approach that enhances the predictive power of a computational intelligence model by integrating numerous models, such as classifiers and experts, to tackle a particular computational issue. Ensemble learning is primarily utilised to bolster the performance of a model, such as classification, prediction, or function approximation, by minimising the possibility of selecting an ineffective model. This technique can also be employed to assign a degree of confidence to the model's decisions, as well as identify the most useful features, fuse data, and implement incremental, non-stationary, and

error-correcting learning strategies. While this article concentrates on classification-related applications, the fundamental ideas outlined can be seamlessly applied to function approximation or prediction problems.

We used the Averaging ensemble approach which includes base models such as Linear Regression, Xgboost and Random forest regressor.

Ensemble Algorithm Used:

- Split the training dataset into train, test and validation dataset.
- Fit all the base models using train dataset.
- Make predictions on validation and test dataset.
- Final prediction is considered as average of base model predictions.

Results

LINEAR REGRESSION

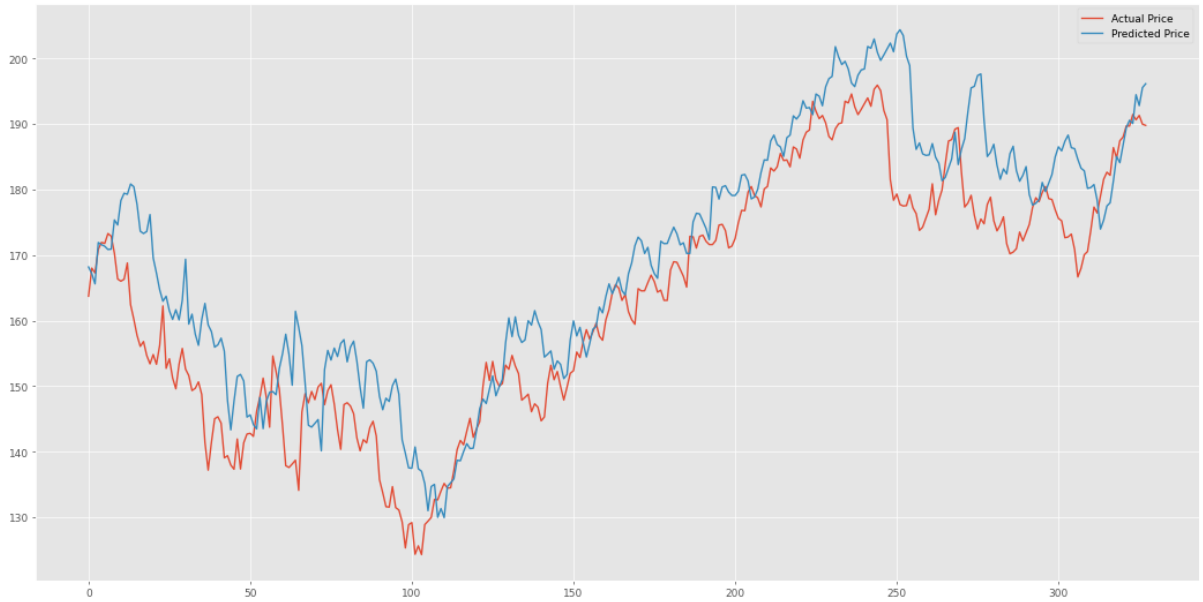
No of days to be forecasted in future = 7

Lag Feature shifted the data by 7 rows .

converted the data into numpy arrays and discarded the last 35 rows.

Apple

GRAPH:



Closing Price Prediction by Linear Regression for next 7 days = [[197.90969873], [197.88891105], [199.71783076], [198.87611194], [199.57234869], [198.17987519], [197.99281775]]

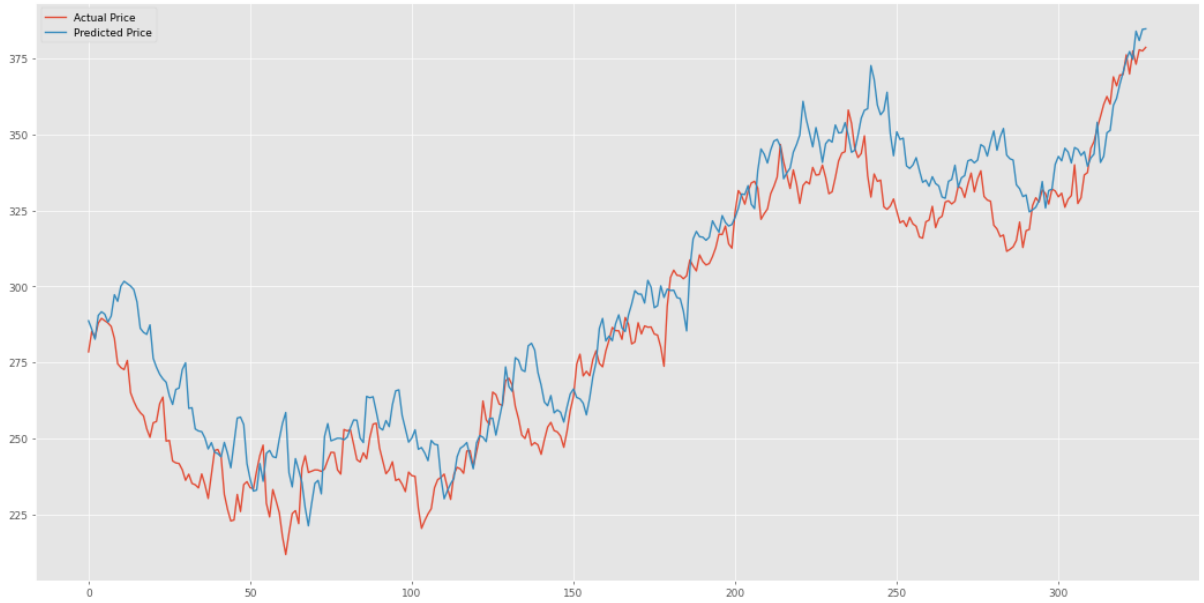
Mean: 198.59108487379885

Mean Square Error: 9.514123965041808

Mean Absolute Error: 7.502394547996277

Microsoft

GRAPH:



Closing Price Prediction by Linear Regression for next 7 days = [[391.43809266], [384.89408744], [392.75309347], [388.22821409], [393.17762876], [392.74272893], [393.96454369]]

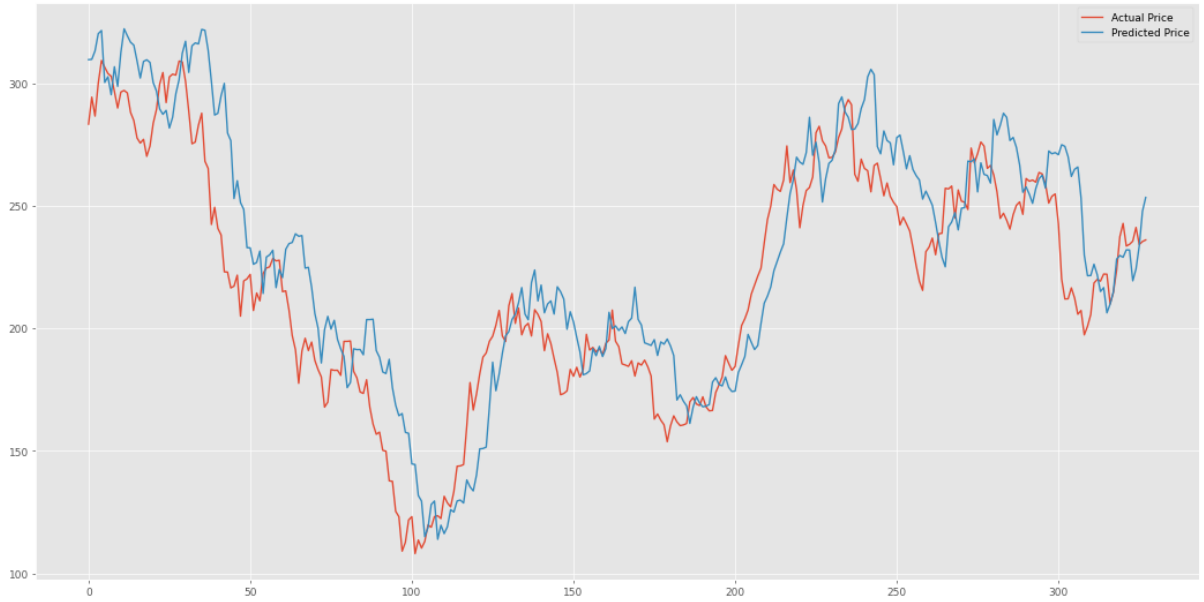
Mean: 391.0283412925106

Mean Square Error: 15.72471088591458

Mean Absolute Error: 12.56621708478854

Tesla

GRAPH:



Closing Price Prediction by Linear Regression for next 7 days = [[243.85007957], [244.58511234], [245.93093743], [251.7283147], [244.49194385], [245.7756408], [246.42785187]]

Mean: 246.11284008025066

Mean Square Error: 24.8201127123811

Mean Absolute Error: 19.877519472973106

Observation

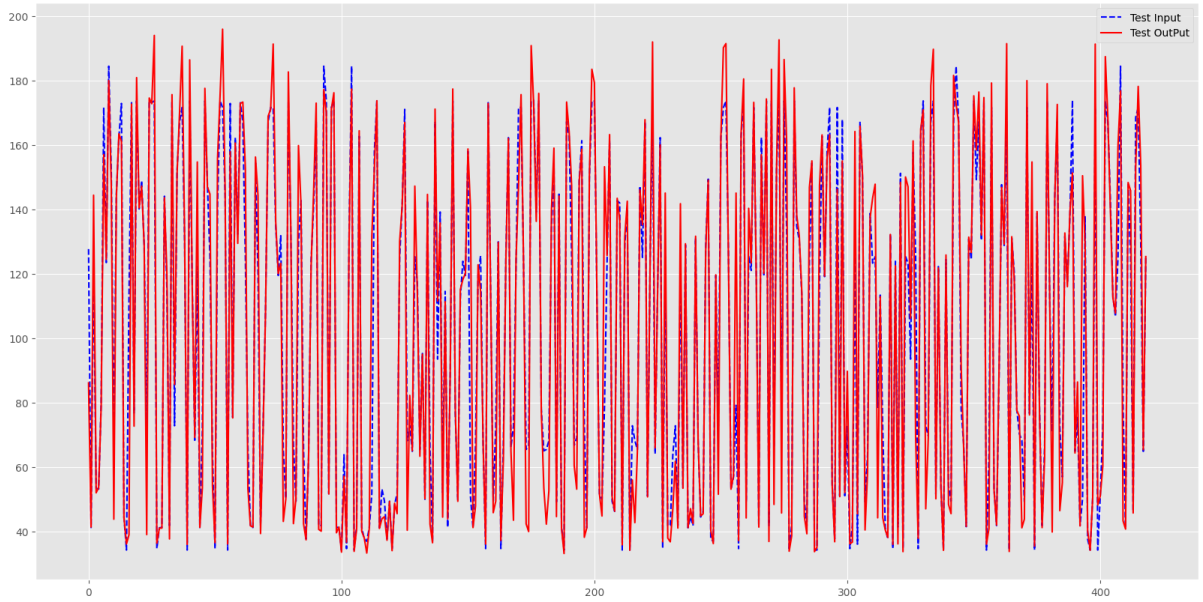
Linear regression forecasts for the next 7 days show a rising trend for Tesla (mean: $246.11, MSE: 24.82, MAE: 19.88), a slightly decreasing trend for Microsoft (mean: $391.03, MSE: 15.72, MAE: 12.57), and a stable trend for Apple (mean: $198.59, MSE: 9.51, MAE: 7.50). While these models offer valuable insights, it is crucial to acknowledge the inherent limitations of linear regression and consider external factors influencing stock prices in real-world scenarios. Continuous monitoring and adjustments to the models may be necessary to enhance predictive accuracy in dynamic financial markets.

XGBOOST

Correlation is a statistical measure that expresses the extent to which two variables are linearly related.

Apple

GRAPH:



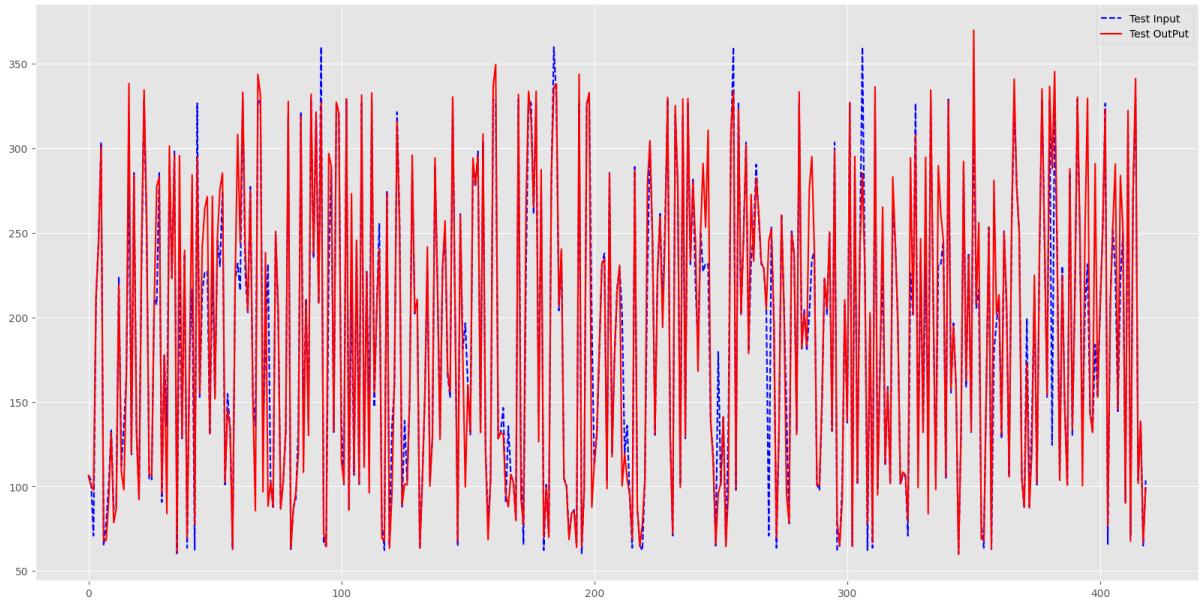
Correlation Coefficients :[[1. , 0.92631552], [0.92631552, 1.]]

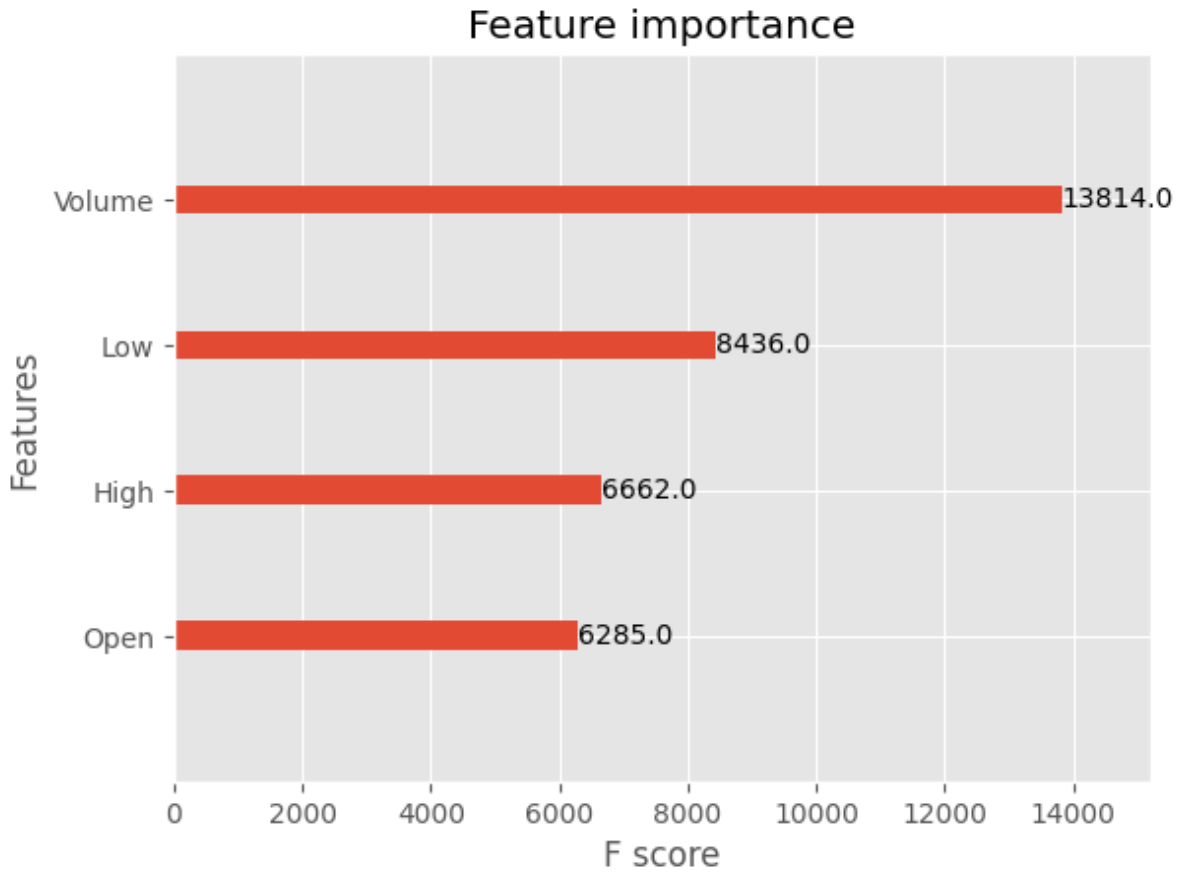
Mean Squared Error (MSE): 374725.1319087832

Mean Absolute Error (MAE): 535.8117800601058

Microsoft

GRAPH:





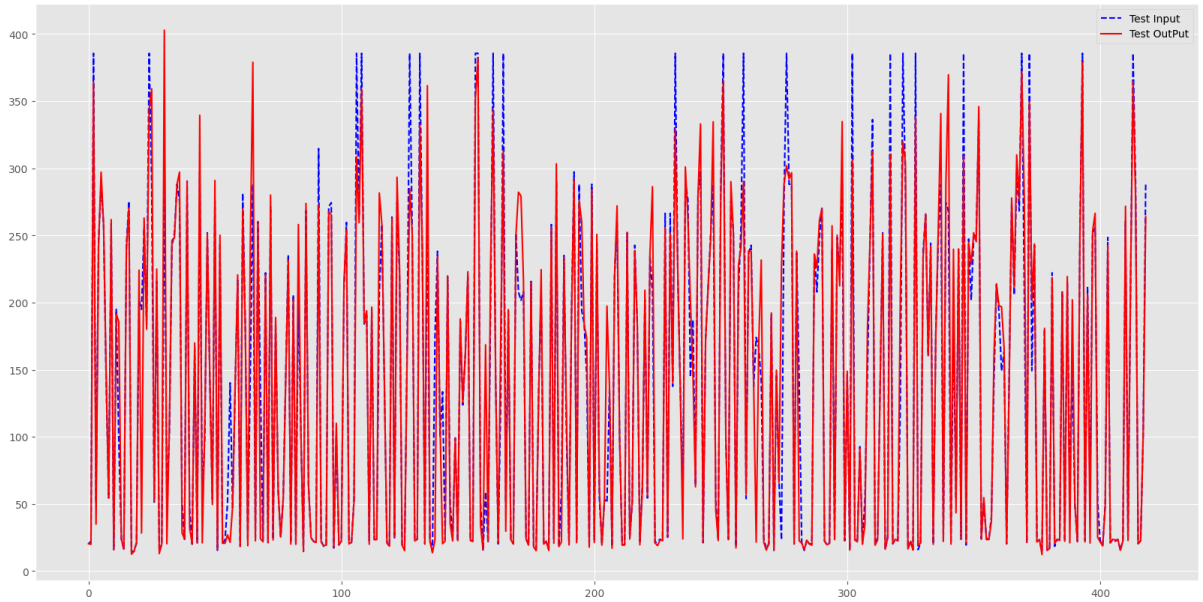
Correlation Coefficients :[[1. , 0.94291134], [0.94291134, 1.]]

Mean Squared Error (MSE): 226987.8715483373

Mean Absolute Error (MAE): 404.0754026444829

Tesla

GRAPH:



Correlation Coefficients :[[1. , 0.92534753], [0.92534753, 1.]]

Mean Squared Error (MSE): 268131.6038969178

Mean Absolute Error (MAE): 462.58977114556797

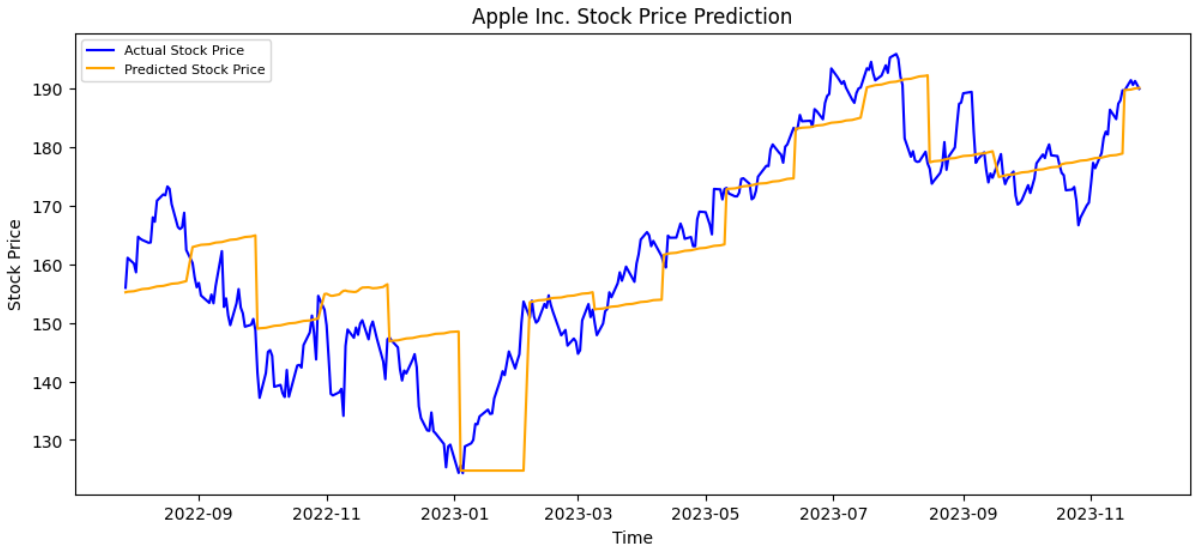
Observation

The high positive correlation coefficients with unspecified variables for all three companies suggest that these variables are influential in predicting the performance or behaviour of the respective companies according to the XGBoost model.

ARIMA-GARCH

Apple:

GRAPH:



MSE: 72.13757099926735

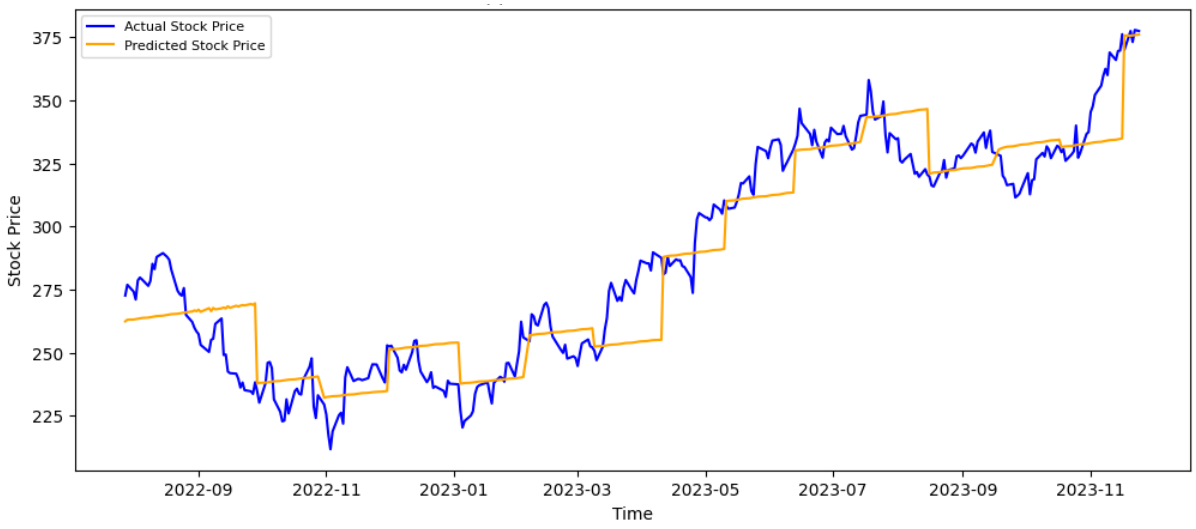
MAE: 6.608010260078055

RMSE: 8.493383954541756

MAPE: 0.04235591471064565

MICROSOFT:

GRAPH:



MSE: 198.02731910918052

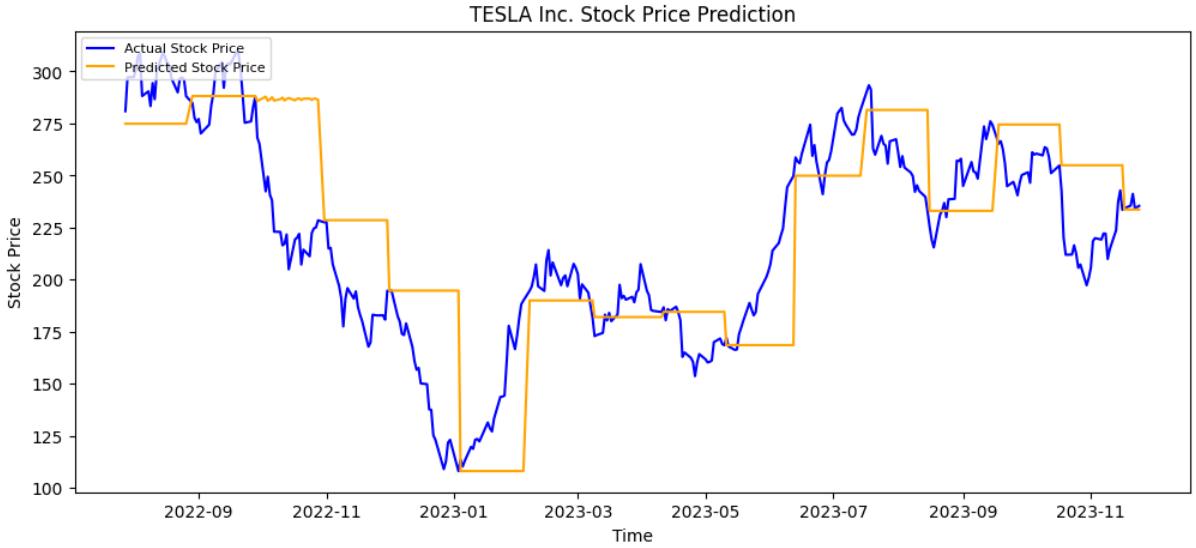
MAE: 11.108213628229043

RMSE: 14.072217988262565

MAPE: 0.03961869904827092

TESLA:

GRAPH:



MSE: 1058.5996931933494

MAE: 25.306525836231355

RMSE: 32.53612904439232

MAPE: 0.12018040766648877

Observation:

It can be seen that ARIMA-GARCH is not a good model to predict stock prices even in the short run without external exogenous variables like volume, volatility or some other market conditions. Here the stock Price of Apple is predicted best. The assumption that Stock price is Autoregressive over long period is false.

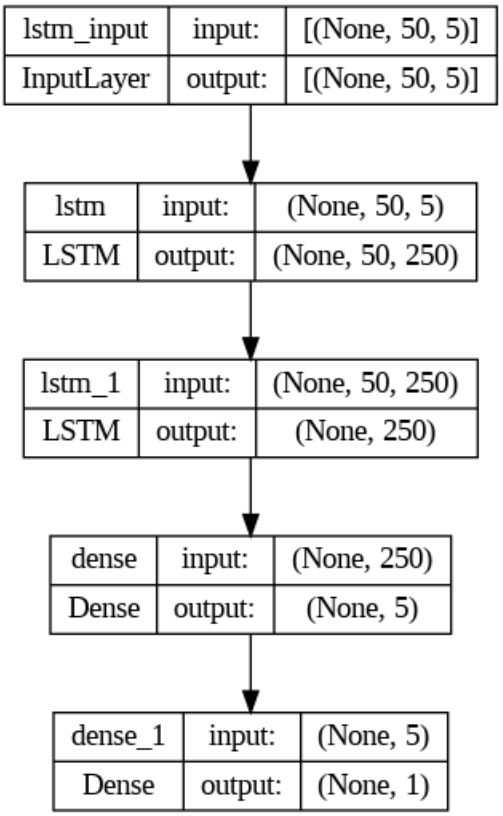
LSTM (Long Short-Term Memory) Model

Model Summary:

Model: "sequential"

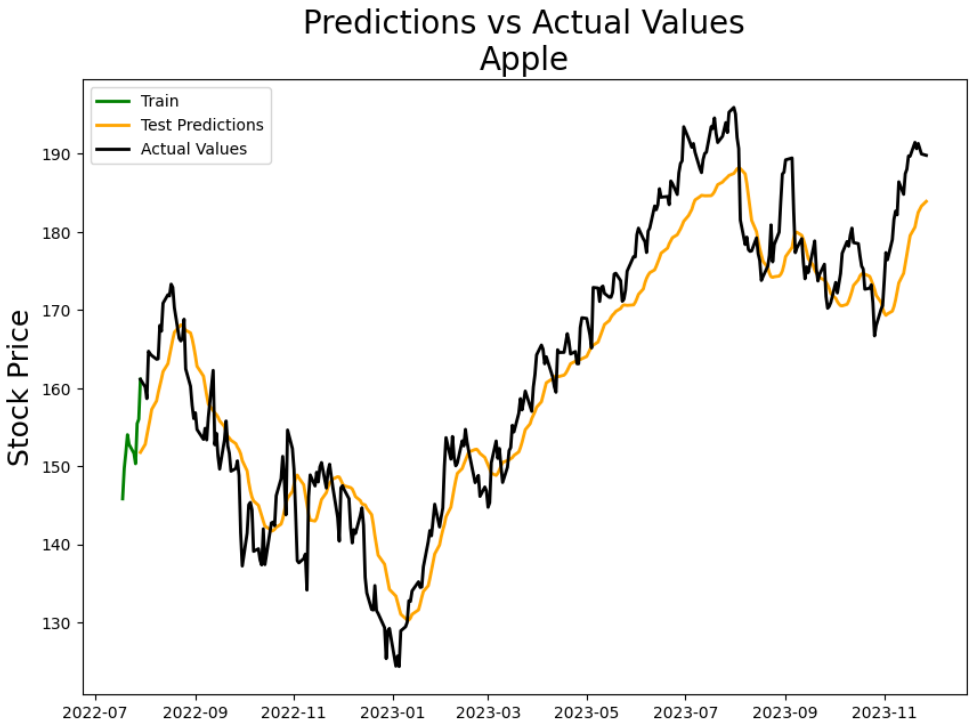
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50, 250)	256000
lstm_1 (LSTM)	(None, 250)	501000
dense (Dense)	(None, 5)	1255
dense_1 (Dense)	(None, 1)	6

Total params: 758,261
Trainable params: 758,261
Non-trainable params: 0



Apple:

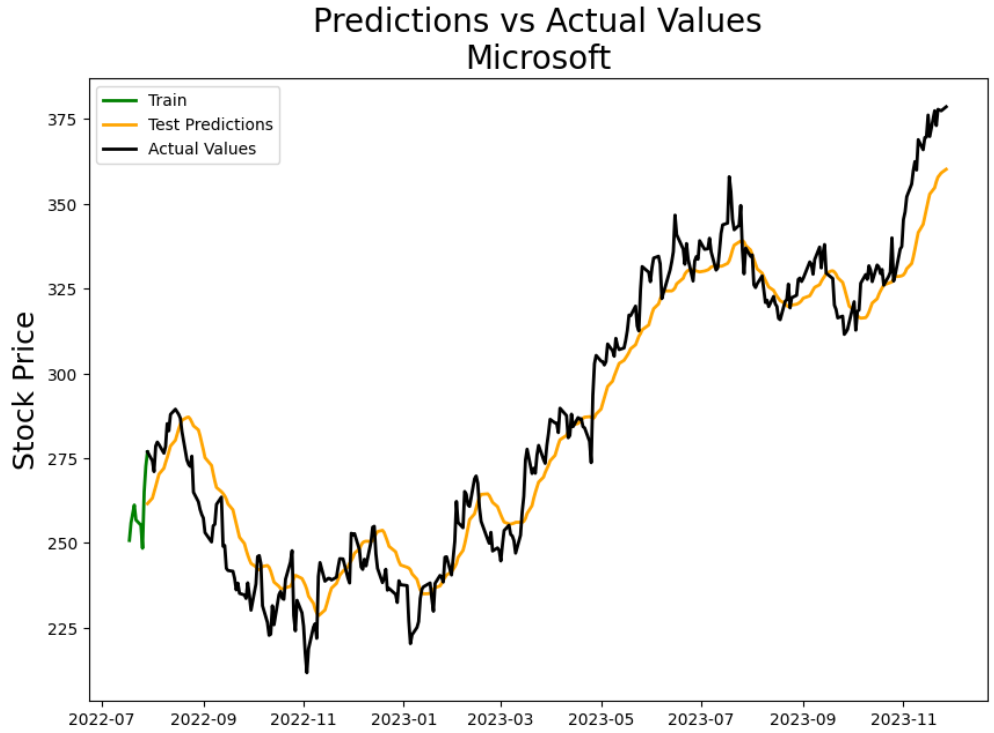
GRAPH:



Mean Absolute Error (MAE): 4.94
Mean Square Error (MSE): 33.97
Root Mean Square Error (RMSE): 5.83
Mean Absolute Percentage Error (MAPE): 3.03 %

Median Absolute Percentage Error (MDAPE): 2.87 %

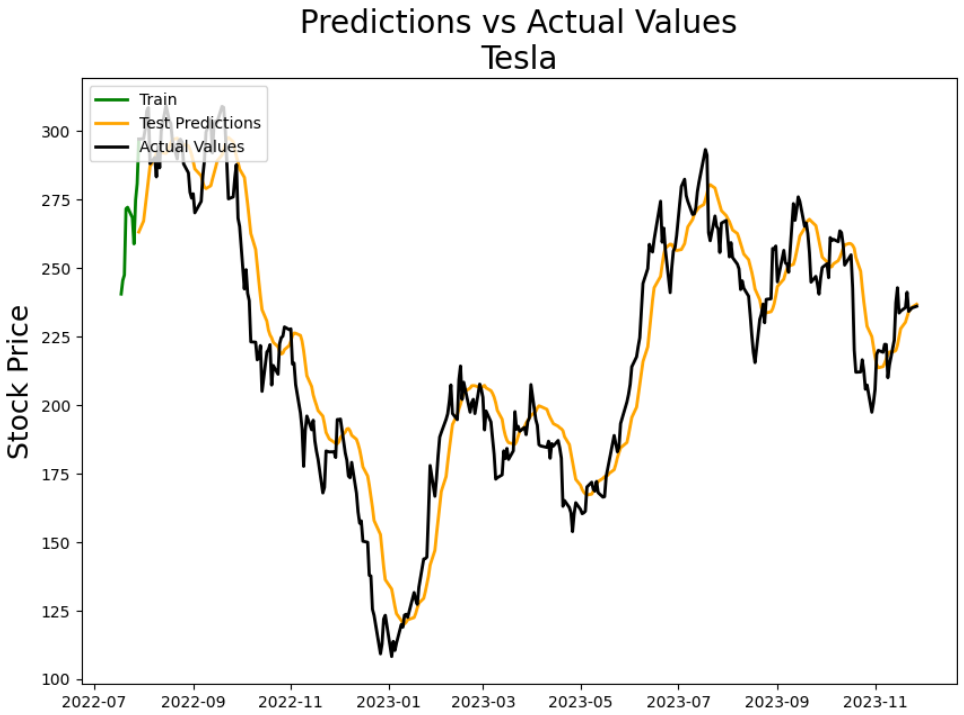
Microsoft:
GRAPH:



Mean Absolute Error (MAE): 8.95
Mean Square Error (MSE): 118.41
Root Mean Square Error (RMSE): 10.88
Mean Absolute Percentage Error (MAPE): 3.18 %
Median Absolute Percentage Error (MDAPE): 2.77 %

Tesla:

GRAPH:



Mean Absolute Error (MAE): 13.14
Mean Square Error (MSE): 268.11
Root Mean Square Error (RMSE): 16.37
Mean Absolute Percentage Error (MAPE): 6.45 %
Median Absolute Percentage Error (MDAPE): 4.76 %

Observation:

On running the same model with the same parameters on three different dataset, we can see that the best prediction is made for the stock price of Apple. The possible reason for this may be less irregularity in the stock price before predictions.

Ensemble

Apple

GRAPH:

MAE: 5.827347598771882
MSE: 48.255737981127304
RMSE: 6.946635011365381
R-Squared: 0.8553719823975824

Microsoft

GRAPH:

MAE: 10.670168140044899
MSE: 177.00556591040043
RMSE: 13.304343873727875
R-Squared: 0.9070265711195944

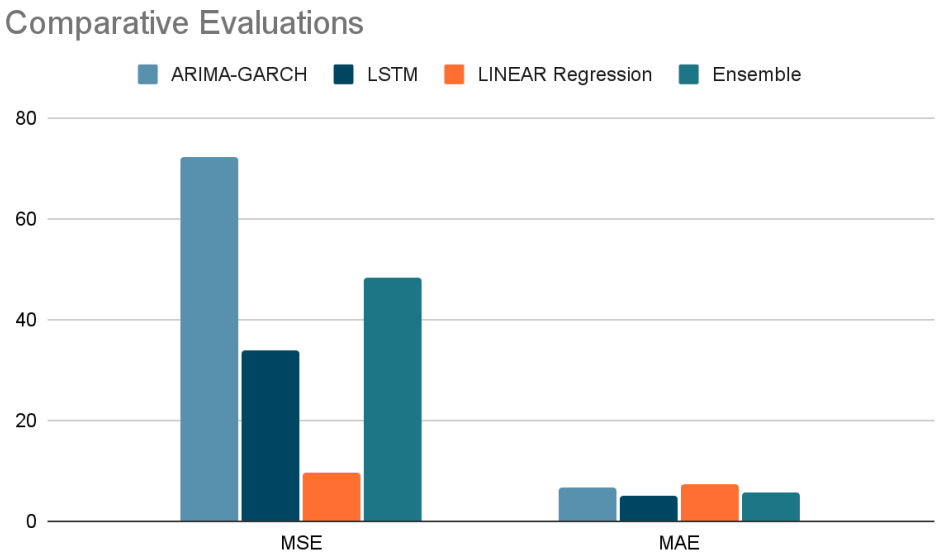
Tesla

GRAPH:

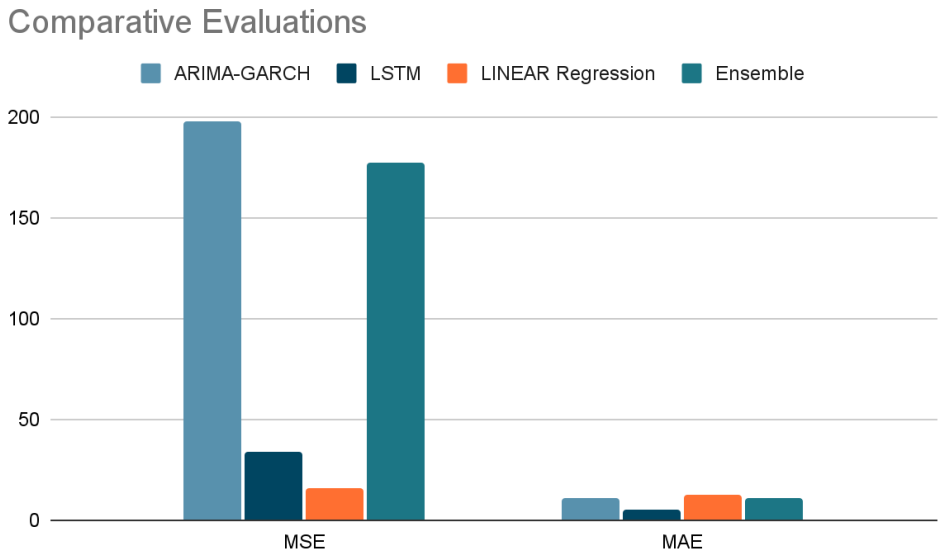
MAE: 17.778058426824867
MSE: 479.1243506500598
RMSE: 21.88890930699974
R-Squared: 0.7907341015162999

Results:

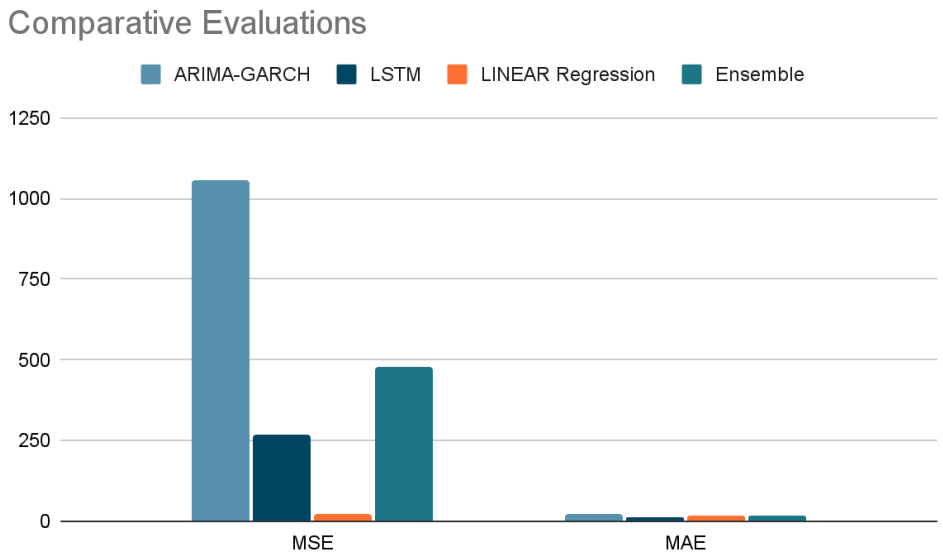
Apple Inc



Microsoft Inc



Tesla Inc



As we can see LSTM appears to be best fit to model stock prices and their volatility. ARIMA seems to be the worst model due to long run future predictions compared to other models. ARIMA is used for monthly forecasts while other models find at most 7 days of future values. After LSTM ensemble appears to be the second best option.

Link to Code:

1. Linear Regression & XGBoost
    a. Tesla
    b. Microsoft
    c. Apple
2. ARIMA-GARCH
    a. Tesla
    b. Microsoft
    c. Apple
3. LSTM
    a. Tesla
    b. Microsoft
    c. Apple
4. Ensemble Learning
    a. Tesla
    b. Microsoft
    c. Apple

References:

1. https://finance.yahoo.com/
2. https://github.com/ranaroussi/yfinance
3. https://arxiv.org/pdf/1204.0375.pdf
4. https://arxiv.org/pdf/2202.03156.pdf
5. https://www.analyticsvidhya.com/blog/2021/07/stock-market-forecasting-using-time-series-analysis-with-arima-model/
6. (PDF) Stock Market Prediction Using Machine Learning(ML)Algorithms (researchgate.net)
7. https://www.mdpi.com/2079-9292/10/21/2717