Stock Forecasting

Rupali Shah | Springboard | Oct 2, 2019

Overview

There are three factors that drive the stock market:

- Fundamental factors drive stock prices based on a company's earnings and profitability from producing and selling goods and services.
- Technical factors relate to a stock's price history in the market pertaining to chart patterns, momentum, and behavioral factors of traders and investors.
- Market sentiment refers to the psychology of market participants.

Technical factors and market sentiment often overwhelm the short run. This project uses Technical factors to predict the stock prices of Oil and Gas companies such as Chevron Corp (CVX) and Exxon Mobil (XON) based on the Stock Price History and the Inflation Rate.

Time Series forecasting is used to predict the stock prices. Depending on the frequency, a time series can be of yearly (ex: annual budget), quarterly (ex: expenses), monthly (ex: air traffic), weekly (ex: sales qty), daily (ex: weather), hourly (ex: stocks price), minutes (ex: inbound calls in a call canter) and even seconds wise (ex: web traffic).

The data was fitted to three different forecasting models using a Daily frequency:

- ARIMA / Auto Arima, and LSTM models were built based on stock price history.
- The VAR model was built using stock price history and inflation rate.

Data Source

The daily stock data for five years (2013 -2017) for Chevron Corporation (CVX) was fetched from the Quandl API.

The 5 year breakdown inflation rate was downloaded from the FRED website as a CSV file.

Exploratory Data Analysis

The Stock Price data from Quandl has 13 columns and 1090 rows with Date as the Index column.

- The Date column is the day on which the stock was traded.
- The columns Open and Close represent the starting and final price at which the stock traded on a day.
- High and Low represent the maximum and minimum price of the share for that day.
- Volume is the total number of shares traded in the day.
- The Split column shows if the stock has been split indicating that the number of shares has increased, and the value of each share has decreased.
- The Adjusted Closing price reflects the stock's value after accounting for any corporate actions. e.g. the closing price adjusted after posting a dividend. Thus, it accounts for the newly reduced value caused by the dividend.

Profit/Loss is determined by the Closing Price. Since Closing Price is the raw price, the Adjusted Closing price was considered as the target variable for this project.

The Inflation Rate dataset from FRED has two columns and 1303 rows.

- The Date column is the date on which inflation rate is recorded.
- The TYSIE column is the inflation rate on a day.

Data Wrangling

Inflation Rate Dataset

- The Date column was converted to Datetime, and indexed.
- The Inflation Rate column was converted to numeric.
- The column names were renamed to make them consistent with the columns of the Stock Price dataset.
- The missing values for Inflation Rate were interpolated (filled by averaging the values of the previous and the next day).

The Stock Price and the Inflation Rate datasets were merged. The new dataset has 3 columns (Date, Adjusted Closing Price, Inflation Rate) and 1090 records.

Models

ARIMA (Auto Regressive Integrated Moving Average)

ARIMA is a popular statistical method which needs a stationary univariate series, and is characterized by three terms: p, d, q where,

- p is the Auto Regressive term, which refers to the past values for forecasting the next value. This value is determined by the PACF plot. 'Auto Regressive' means it is a Linear Regression model. Since Linear regression models work best when the predictors are not correlated and are independent of each other, the series needs to be stationary.
- d is the minimum number of differencing needed to make the series stationary.
- q is the Moving Average term, which defines the number of past forecast errors, and is determined using the ACF plot.

ARIMA model equation:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

ARIMA model in words:

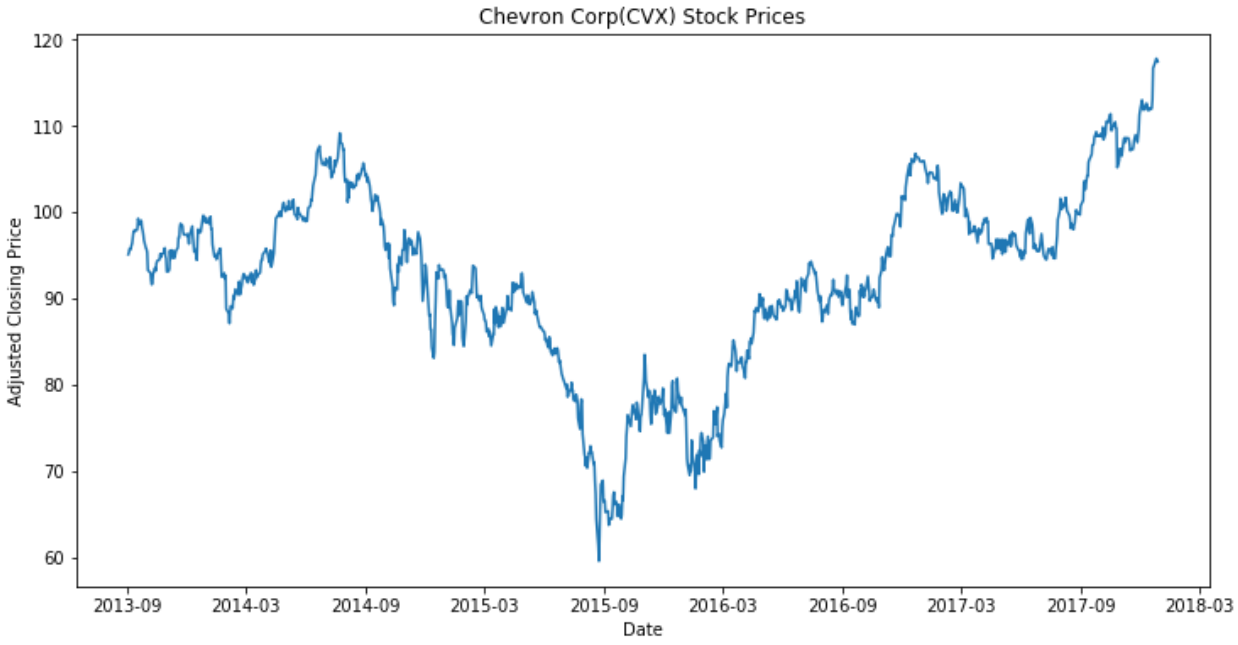
Predicted Yt = Constant + Linear combination Lags of Y (up to p lags) + Linear Combination of Lagged forecast errors (up to q lags)

Data Preprocessing

Value of d

Stationarity Check

- Visual
  The plot below indicates that the Time Series is neither seasonal nor stationary.



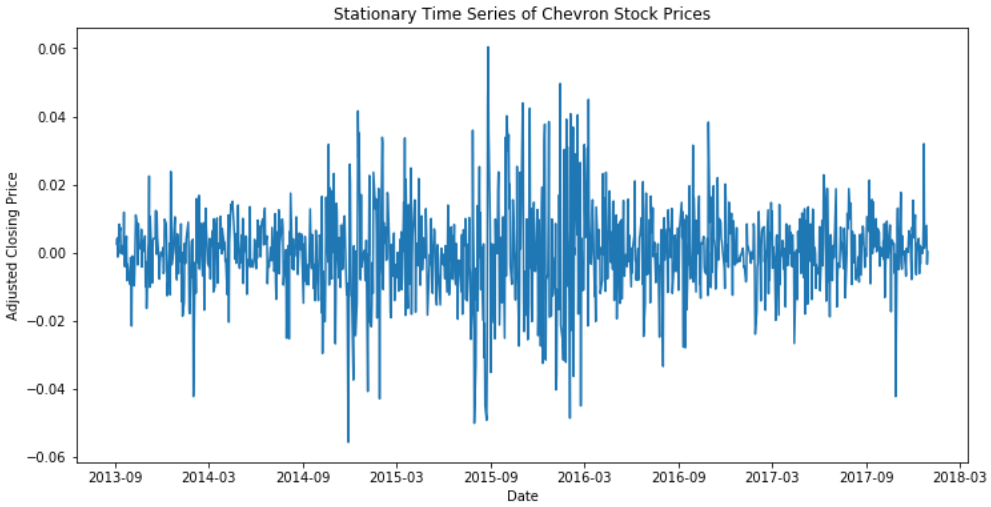
- Statistical Test

  Augmented-Dickey Fuller(ADF) test

  The null hypothesis of the ADF test is that the time series is non-stationary. If the p-value of the test is less than the significance level (0.05), then the null hypothesis is rejected, and it is inferred that the time series is indeed stationary.

  The ADF test results gave a p-value of 0.7 suggesting that the Time Series was not stationary.

  The Time Series was made stationary by differencing it. This was done by first transforming the data to a logarithmic scale to distribute the data normally, and then subtracting the previous value form the current value.



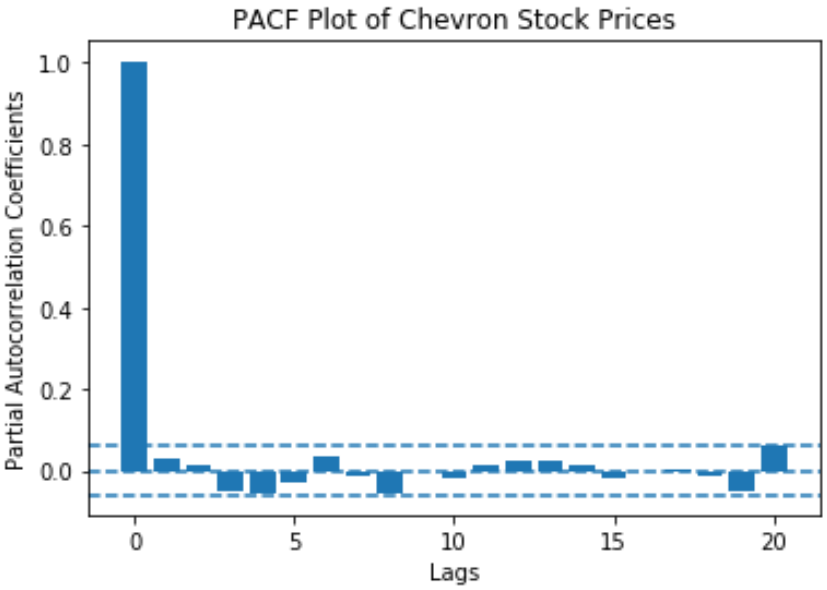
The ADF-test results after differencing suggested that the Time Series was stationary since p-value of 0.3 was less than the significance level.

The Time Series was differenced once, so the order of differencing is 1.

Value of p

The order of the AR term(p) is determined by inspecting the PACF plot. Partial auto correlation is the correlation between the series and its lags. Any autocorrelation in a stationary series can be rectified by adding enough AR terms. The order of AR term is equal to the lags that cross the significance limit in the PACF plot.

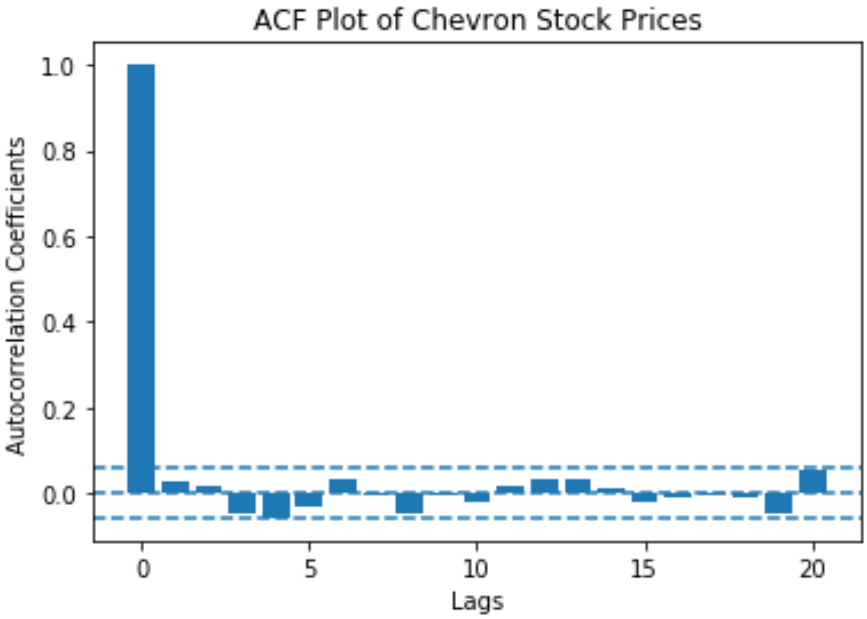
In the plot below p = 0.



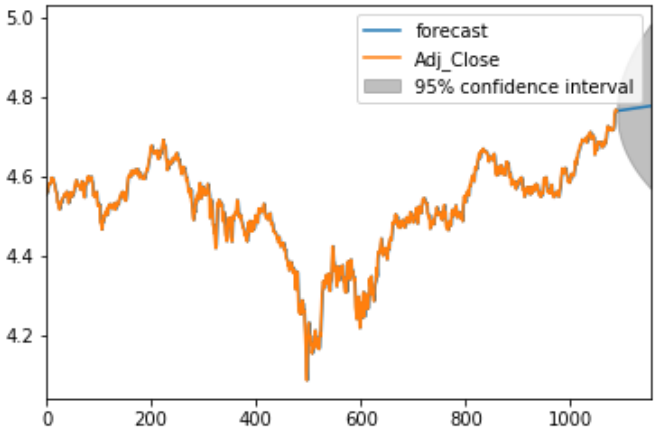
Value of q

The order of the MA term(q) is determined by inspecting the ACF plot. The ACF plot tells how many MA terms are required to remove any autocorrelation in a stationary series. An MA term is the error in the lagged forecast.

Value of q = 0.



The model was built with the combination of (p,d,q) = (0,1,0)



AUTO ARIMA

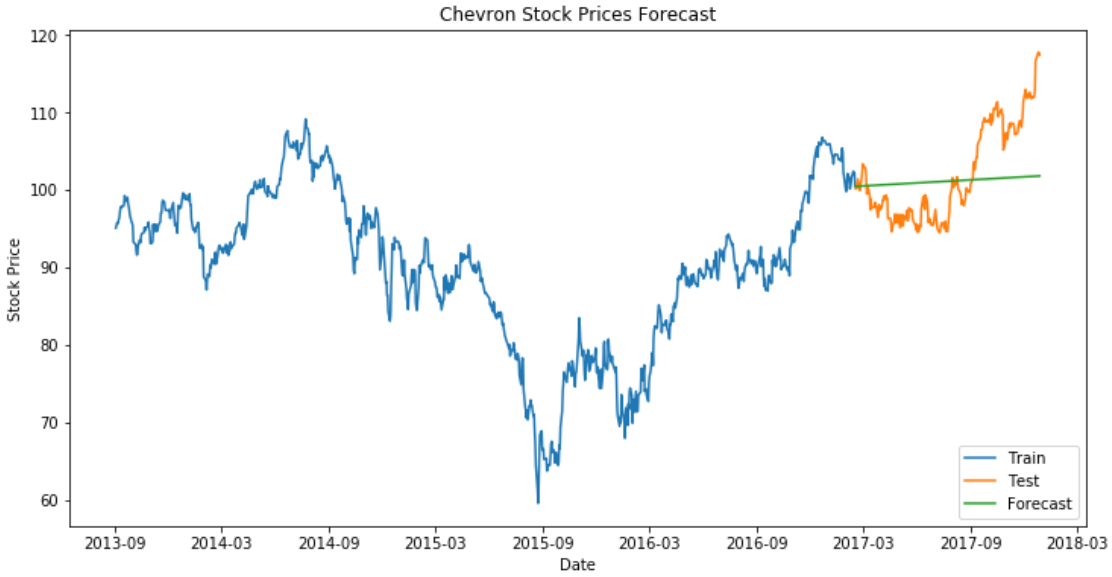
Hyperparameter tuning for the ARIMA model can be quite time consuming. Auto ARIMA automatically selects the best combination of (p,q,d) that provides the least error. Auto ARIMA considers the AIC and BIC values generated to determine the best combination of parameters. AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) values are estimators to compare models. The lower these values, the better the model.

The Stock Price data was split into Train and Test keeping the Time Series intact. The first 80% of the records were saved as the Training set, and the remaining 20% as the Testing set. The train_test_split and k-fold validation cannot be used here, since it can disrupt the pattern in the Time Series.

The training data was then fitted to the AUTO ARIMA model from the Pyramid library. The model selected the best combination (0,1,0) based on the AIC and BIC values.

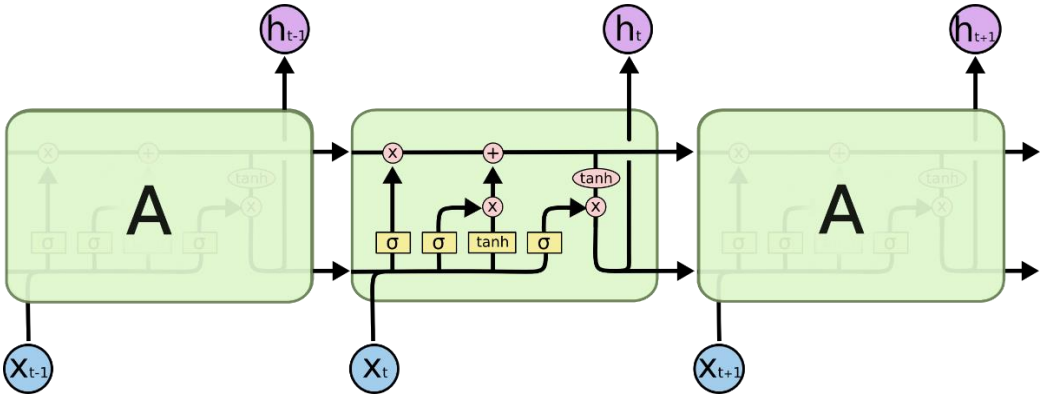
The predictions were made on the test dataset.

This model gave a Root Mean Squared Error of 5.92.



LSTM (Long Short Term Memory)

LSTMs are a kind of RNN (Recurrent Neural Network) that are capable of learning long-term dependencies. LSTMs have the form of a chain of repeating modules of neural network. The repeating module contains four interacting layers. The LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell. The activation function of the LSTM *gates* is often the logistic sigmoid function.



Data Preprocessing

- The Stock Price dataset was split into Train and Test sets.
- The Training set was converted to a NumPy array and flattened to one column.
- The Training data was normalized in the range of 0 to 1 using MinMaxScaler.
- Datasets X_train and y_train were created from the Training set with 30 timesteps and 1 output, and converted to NumPy arrays.
- X_train was reshaped into a 3D array, with the number of X_train samples, 30 timesteps, and one feature at each step. This is the input shape required for a 3D tenor.

Model Building

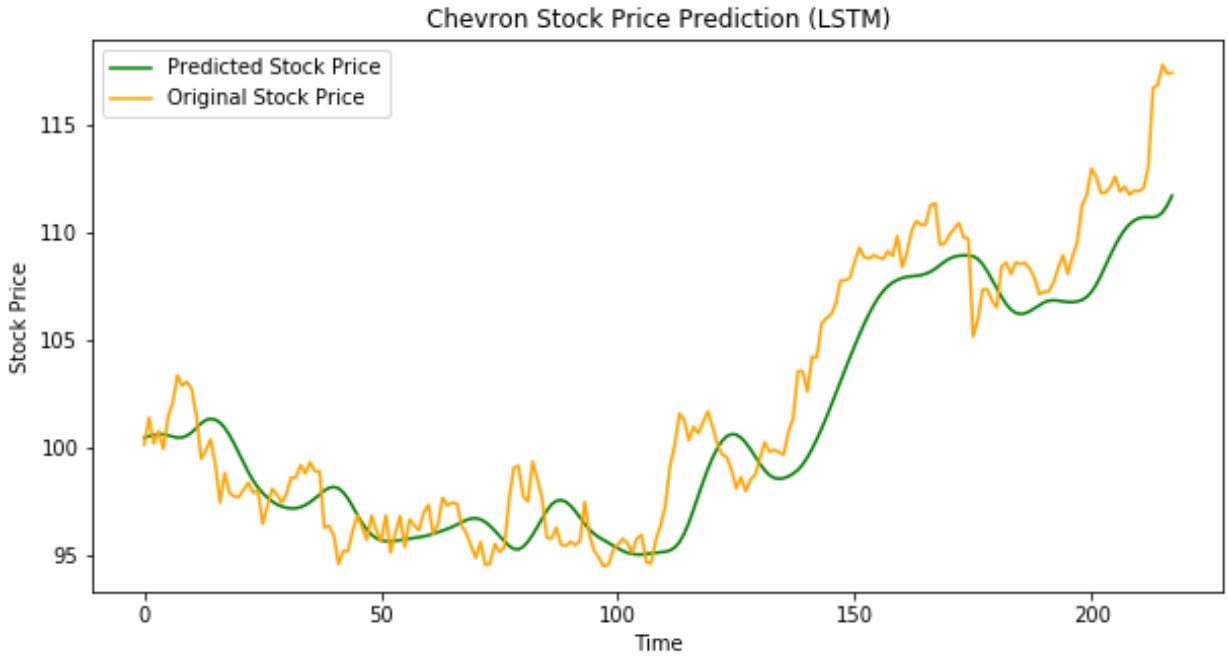
A sequential model which is a linear stack of layers was used. 4 LSTM layers with 50 units that returned sequences were added. A dropout layer was added to each LSTM layer. A fully connected Dense layer was added at the end with 1 unit.

The model was compiled using the 'adam' optimizer, and trained for 24 epochs with a batch size of 32.

Forecasting

Predictions were made by concatenating the Train and the Test datasets. The last 30 records of the Train dataset and all records of the Test dataset were used as inputs. The input data was converted to a NumPy array, reshaped to one column and scaled using MinMaxScaler. The input data was reshaped to a 3D array with test samples, 30 timesteps, and one feature. Predictions were made on the 3D input. The predicted data was inverse transformed to normal values.

The model gave an RMSE of 2.37



VAR (Vector Auto Regression)

VAR is one of the most commonly used methods for multivariate Time Series forecasting. In a VAR model, each variable is a linear function of the past values of itself and the past values of all the other variables. Unlike AR, VAR can understand and use the relationship between several variables.

Data Preprocessing

Stationarity Check

A stationary time series will often give a better set of predictions. The multivariate series is checked for stationarity based on the Eigen values. This is done using the Coint Johansen test.

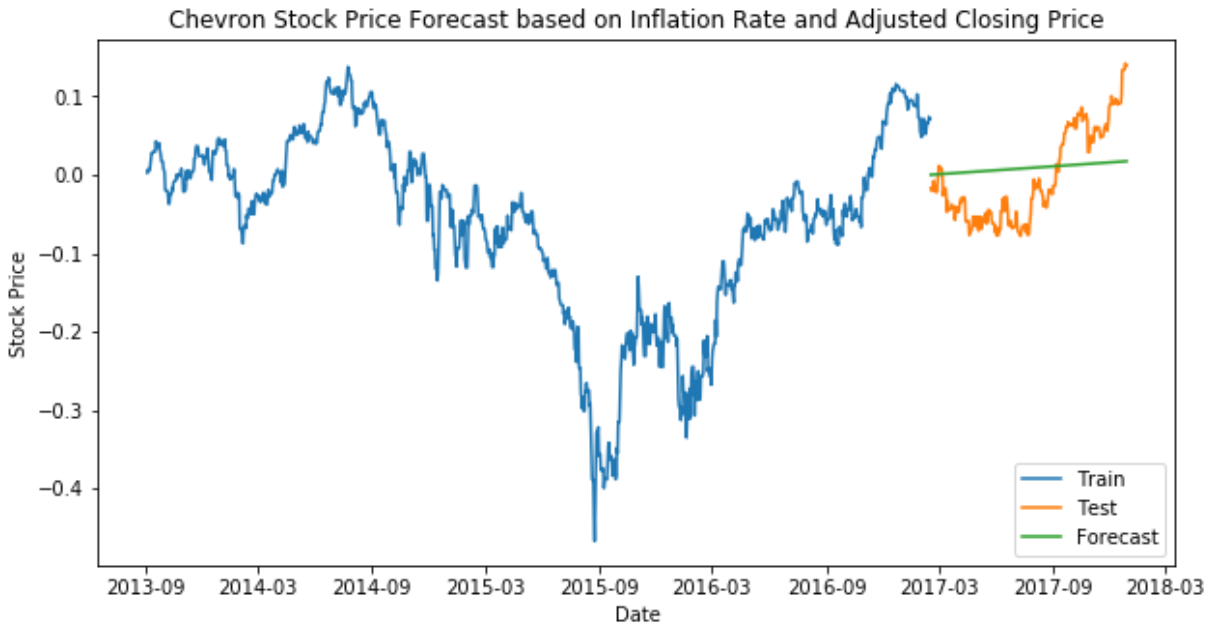
The Time Series was differenced once to make it stationary.

The Time Series is on a 'Business Day' Frequency as there is no trading on weekends. The gap in the series was filled by first converting it to a 'Daily' Frequency, and then filling the values of the stock prices for weekends using a Forward Fill.

Model Building

The dataset was split into Train and Test while accounting for the time component. The Training data was then fit to the VAR model from the statsmodel library, letting the model determine the number of lags.

Predictions are made on the test dataset. The model gave an RMSE of 0.0127



Results

- The auto ARIMA model uses past data to understand the pattern in the time series models, but these predictions are not close to the real values.
- The LSTM model prediction is close to the real data, though it does not predict the exact values.

Model	Auto ARIMA	LSTM	VAR
RMSE	5.92	2.37	0.01

Future Work

- Train the VAR model with additional features such as Twitter sentiment.
- Train the LSTM model with multivariate data.