Time Series Analysis – I

Data Assignment

By Manas Dixit

When it comes to business, time is extremely crucial. Every second is worth money, and every national and global economy is reliant on it. In order to comprehend a variable that varies over time, time series analysis has become an often-used tool in the field of analytics.

Time series data means that data is in a series of particular time periods or intervals. Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time.

Introduction

The goal of this exercise is to examine how the closing price and volume-weighted average price of HDFC's shares have changed over time. It begins with data preparation for visualizations and continues with a detailed exploratory data analysis that includes the influence of COVID-19 on HDFC's stocks, as well as time series model creation and the use of Facebook's new Prophet tool for time series forecasting.

We have used Python for our analysis. The link to the Github with the complete code and dataset can be found here.

About HDFC Bank

The Housing Development Finance Corporation Limited or HDFC was among the first financial institutions in India to receive an "in principle" approval from the Reserve Bank of India (RBI) to set up a bank in the private sector. This was done as part of RBI's policy for liberalization of the Indian banking industry in 1994.

HDFC Bank was incorporated in August 1994 in the name of HDFC Bank Limited, with its registered office in Mumbai, India. The bank commenced operations as a Scheduled Commercial Bank in January 1995. Currently, it is the biggest private sector bank in India with a market cap of Rs. 8,46,022 crores.

About the dataset

The dataset used in this project is the price history and trading volumes of HDFC (Housing Development Finance Corporation Limited) Bank, a NIFTY 50 stock from NSE (National Stock Exchange) India. The data is from January 1, 2000, to July 31, 2020.

Here is a quick description of the columns in our dataset:

- Date — Date of trade
- Symbol — Name of the company (HDFC Bank)

- Series — We have only one series (EQ). It stands for Equity. In this series, intraday trading is possible in addition to delivery
- Prev Close — Refers to the prior day's final price of a security when the market officially closes for the day
- Open — The open is the starting period of trading on a securities exchange or organized over-the-counter market
- High — Highest price at which a stock traded during the course of the trading day
- Low — Lowest price at which a stock traded during the course of the trading day
- Last — The last price of a stock is just one price to consider when buying or selling shares. The last price is simply the most recent one
- Close — The close is a reference to the end of a trading session in the financial markets when the markets close for the day
- VWAP (Volume-weighted average price) — It is the ratio of the value traded to total volume traded over a particular time horizon. It is a measure of the average price at which a stock is traded over the trading horizon
- Volume — It is the amount of a security that was traded during a given period of time
- Turnover — It is a measure of sellers versus buyers of a particular stock. It is calculated by dividing the daily volume of a stock by the "float" of a stock, which is the number of shares available for sale by the general trading public
- Trades — The number of shares being traded on a given day is called trading volumes
- Deliverable Volume — The quantity of shares which actually move from one set of people (who had those shares in their demat account before today and are selling today) to another set of people (who have purchased those shares)
- % Deliverable — Shares which are actually transferred from one person's demat account to another person's demat account

Now, let us check out the statistical measures of our dataset.

	Prev Close	Open	High	Low	Last	Close	VWAP
count	5097.000000	5097.000000	5097.000000	5097.000000	5097.000000	5097.000000	5097.000000
mean	994.714616	994.963106	1007.384638	981.513302	994.896959	994.892849	994.473841
std	644.547489	644.176437	650.255101	638.193293	644.471667	644.441501	644.235252
min	157.400000	162.150000	167.900000	157.000000	163.000000	163.400000	161.400000
25%	459.700000	460.150000	467.850000	452.650000	460.800000	460.500000	460.780000
50%	885.850000	884.000000	902.000000	868.250000	887.800000	885.900000	884.490000
75%	1406.250000	1405.000000	1425.000000	1378.600000	1406.650000	1406.250000	1403.020000
max	2565.800000	2566.000000	2583.300000	2553.700000	2563.000000	2565.800000	2570.700000

Figure 1: Statistical measures of our dataset

We can see that we have a lot of outliers in our dataset as the max value is close to double the 75th percentile. Also, the standard deviation and other statistical measurements is more or less equal among all the features.

Data Preparation

Before we begin to visualize our data and build and apply models, it is important to cleanse the data. In our dataset, thankfully, we only have a few null values that we need to get rid of.

We will also use the time to extract a few more features to perform in-depth Exploratory Data Analysis (EDA), as can be seen below.

Date	Month	Week	Day	Day of week
2000-01-03	1	1	3	0
2000-01-04	1	1	4	1
2000-01-05	1	1	5	2
2000-01-06	1	1	6	3
2000-01-07	1	1	7	4

Figure 2: We have indexed the dates

Data Visualization

Now we will visualize our data in the form of line charts to understand the trend, seasonality and other notable time series analysis concepts.

- Distribution of stock measures

First, we have the histogram distribution of the stock measures such as open, close, high, low and as well as VWAP.

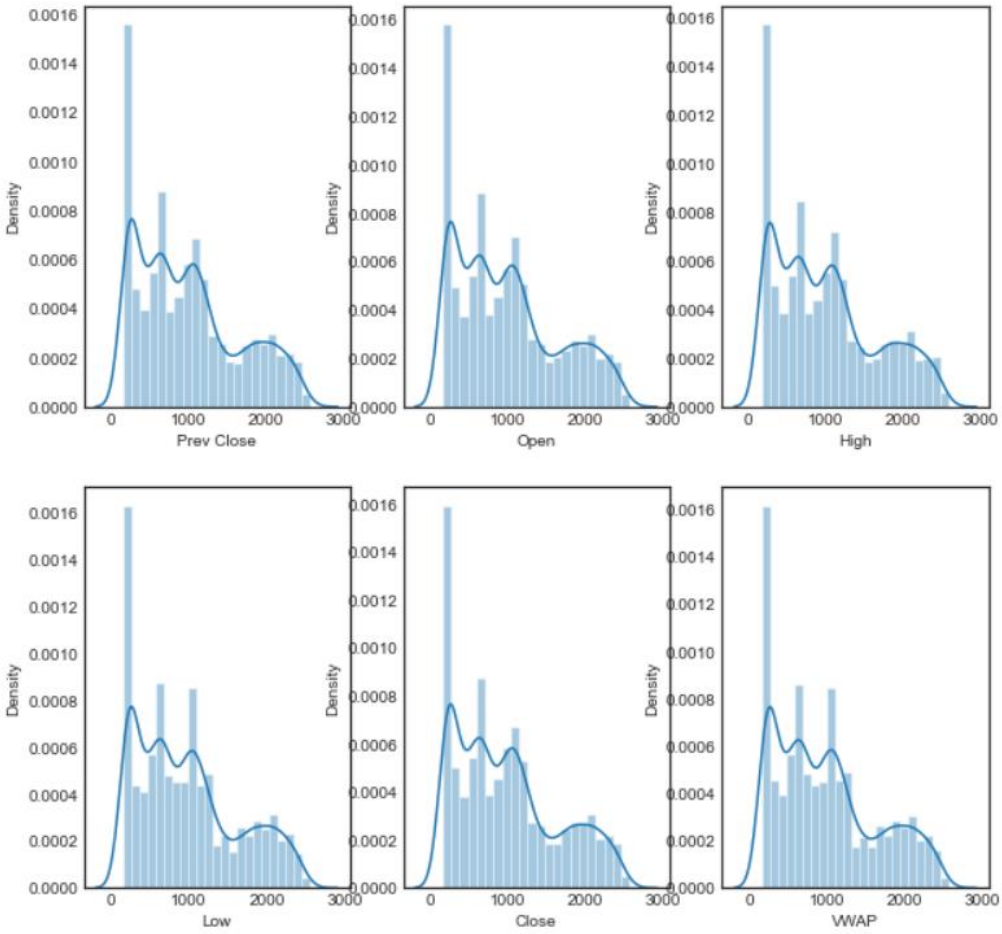

Figure 3: Distribution of stock measures

We can see that all the distributions are right skewed.

- VMAP over time

Figure 4: VMAP over time. Refer to the Github link to interact with the plot.

It is clear from the above figure that there has been a gradual increase in VMAP over time. There are three notable spikes – January 2008, October 2010 and July 2019. Another insight that we can draw from this is that there has been a substantial decline in the VMAP in 2020, particularly after the outbreak of COVID-19, which isn't surprising.

- Uni-variate analysis of Open, Close, High and Low

Figure 5: Uni-variate analysis of Open, Close, High and Low

As we discussed earlier, all these parameters follow a similar pattern without much deviation. There is a break in 2012 and 2020, indicating a sudden dip in the market for the bank.

- Uni-variate analysis of volume of share over the years

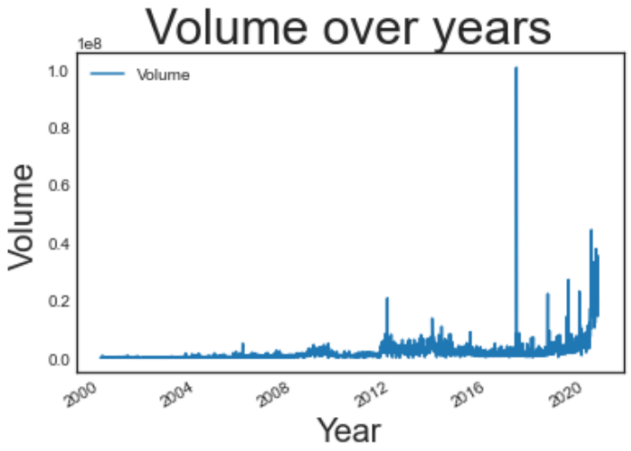

Figure 6: Uni-variate analysis of Volume of share over the years

The volume of trade has grown significantly in the recent years. There was a peak in 2018, as is visible in the figure above.

Moving Average Analysis

Moving average is a smoothing technique applied to time series to remove the fine-grained variation between time steps. The hope of smoothing is to remove noise and better expose the signal of the underlying causal processes. Moving averages are a simple and common type of smoothing used in time series analysis and time series forecasting. Calculating a moving average involves creating a new series where the values are comprised of the average of raw observations in the original time series. A moving average requires that you specify a window size called the window width. This defines the number of raw observations used to calculate the moving average value. The "moving" part in the moving average refers to the fact that the window defined by the window width is slid along the time series to calculate the average values in the new series.

In our case, we consider the moving mean and standard deviation for 3,7 and 30 days.

Volume_std_lag3	Volume_std_lag7	Volume_std_lag30	VWAP_mean_lag3	VWAP_mean_lag7	VWAP_mean_lag30	VWAP_std_lag3	VWAP_std_lag7	VWAP_std_lag30
569111.000000	695677.312500	898106.500000	994.284729	993.944580	992.096436	10.809008	17.445614	38.774673
569111.000000	695677.312500	898106.500000	169.520004	169.520004	169.520004	10.809008	17.445614	38.774673
95778.320312	95778.320312	95778.320312	172.255005	172.255005	172.255005	3.867874	3.867874	3.867874
75766.851562	75766.851562	75766.851562	171.236664	171.236664	171.236664	3.254417	3.254417	3.254417
45964.089844	64369.164062	64369.164062	170.876663	170.537506	170.537506	3.582462	3.002692	3.002692

Figure 7: A glimpse of the lags

We have created the moving average and standard deviation for the respective days across High, Low, Volume, VWAP.

- High vs Low with mean and standard deviation lag — 30 days

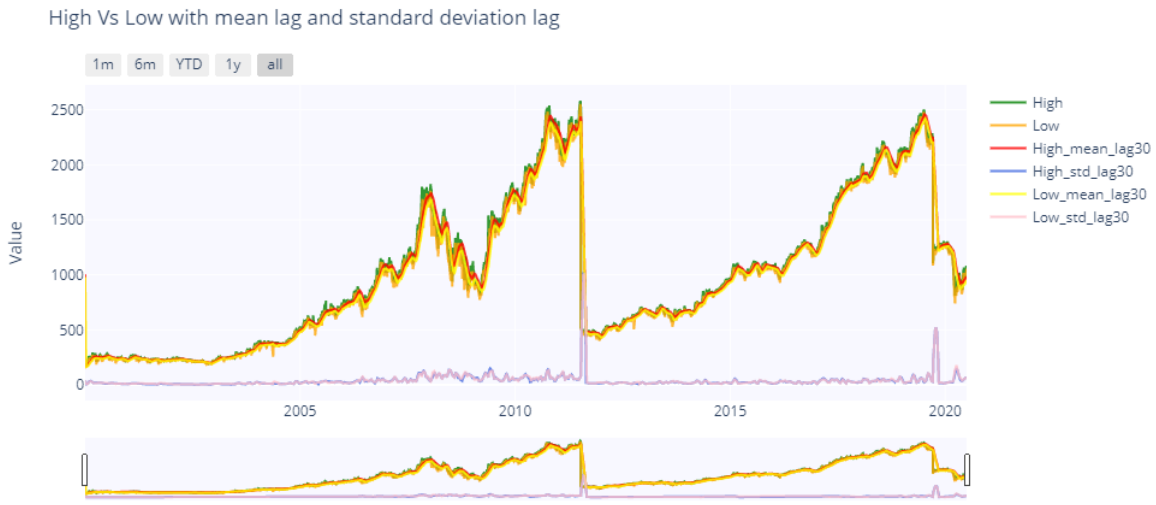

Figure 8: High vs Low with mean and standard deviation lag — 30 days. Refer to the Github link to interact with the plot.

Considering the standard deviation, there's a high deviation whenever there is a drop in the price of stock. We can make use of standard deviation to understand where the company faced loss.

- Volume with mean and standard deviation lag — 30 days

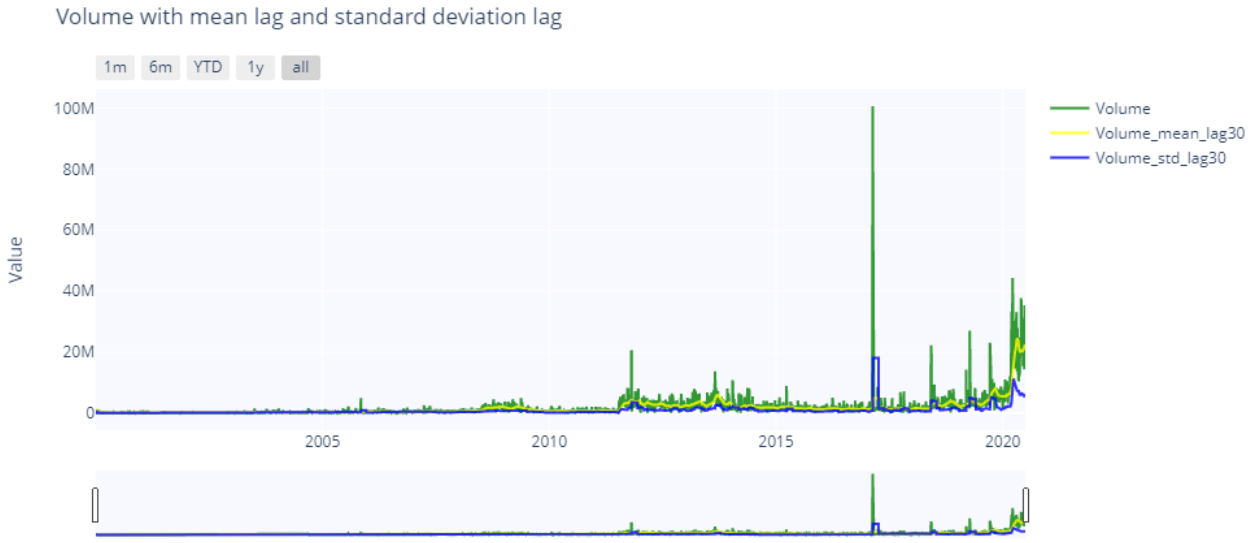

Figure 9: Volume with mean and standard deviation lag — 30 days. Refer to the Github link to interact with the plot.

Here we have a neat representation of the moving average and standard deviation graph. There's a lot of deviation in the volume value in 2020 and the corresponding mean is high compared to standard deviation.

Impact of COVID-19

- Performance after lockdown – VWAP

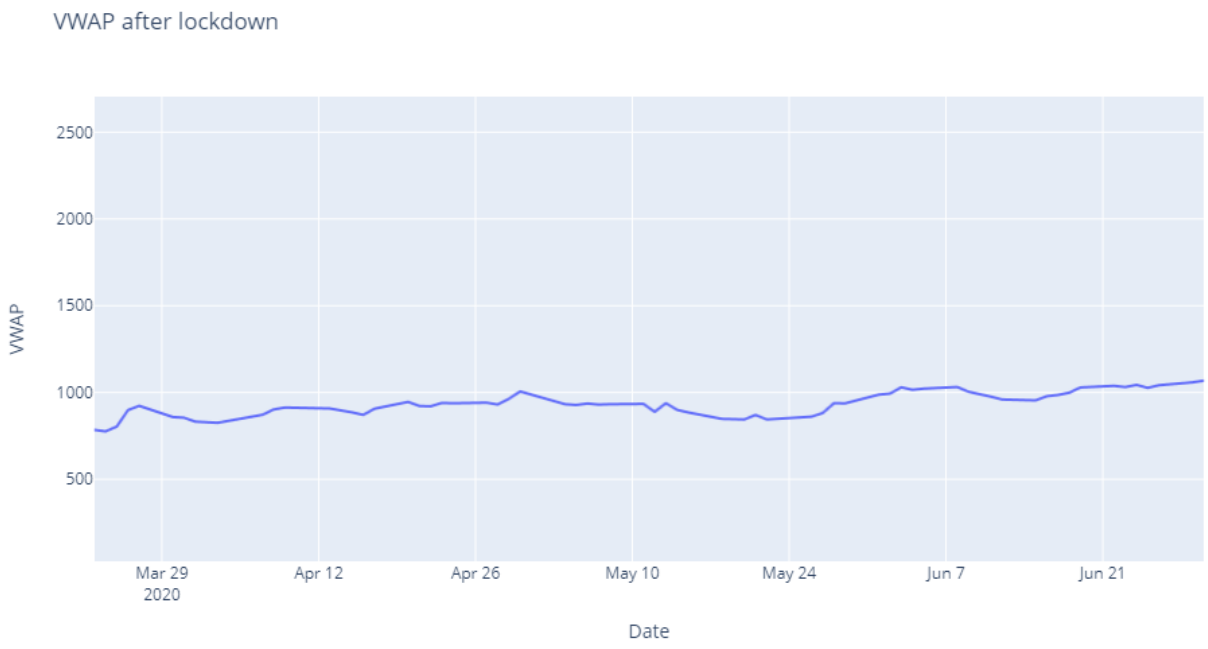

Figure 10: Performance after lockdown – VWAP

At the start of the lockdown the VMAP was between 500 and 1000. By the end of the lockdown, it went just over 1000.

- Candlestick after Lockdown (Open, Close, High, Low)

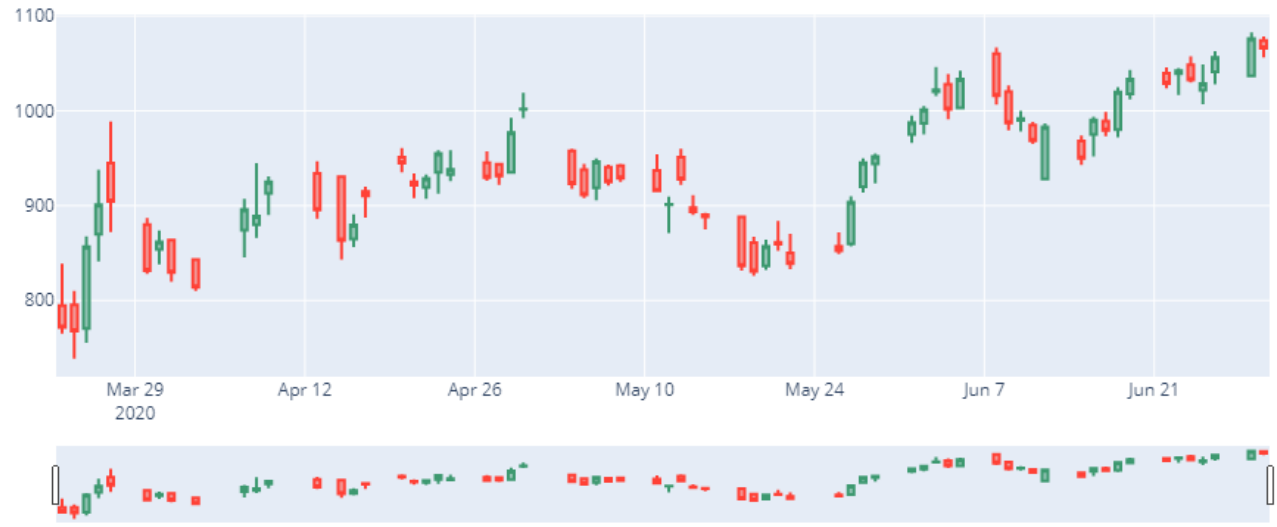

Figure 11: Candlestick after Lockdown

- Volume during Phase 1 Lockdown (25 March — 14 April) and Phase 2 Lockdown (15 April — 3 May)

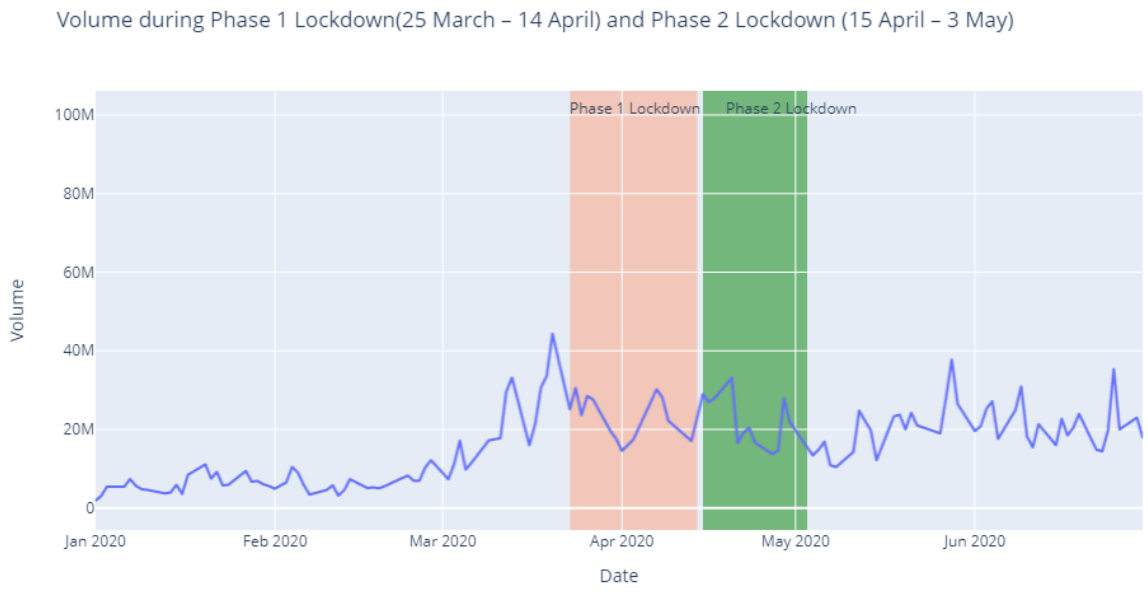

Figure 12: Volume during Phase 1 Lockdown and Phase 2 Lockdown

We see that at the start of the first lockdown, there is a sudden dip, which was experienced by pretty much every company in the stock market. However, the situation did stabilize after the dip.

Stationarity Conversion

A common assumption in many time series techniques is that the data are stationary. A stationary process has the property that the mean, variance and auto-correlation structure do not change over time.

While there are many ways to check the stationarity of a time series, we only look at two ways to check the stationary of our time series.

The first is by looking at the data. By visualizing the data, it should be easy to identify a changing mean or variation in the data. For a more accurate assessment there is the Dickey-Fuller test.

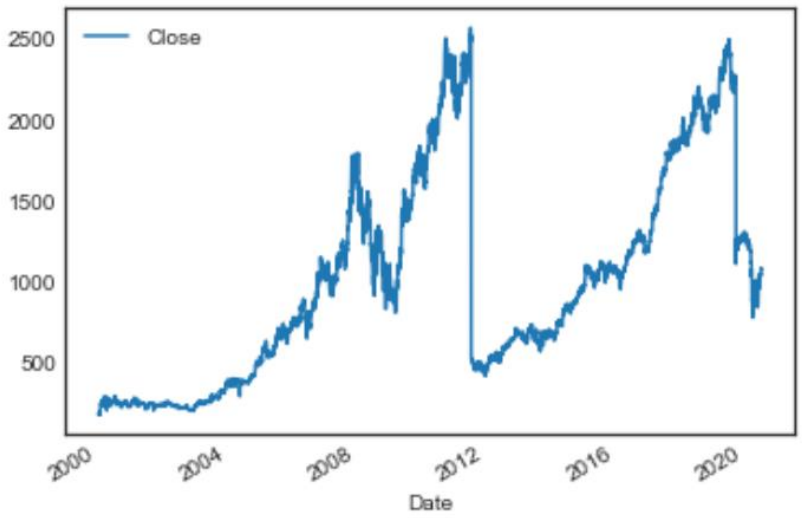

Figure 13: Checking stationarity by visualizing the data

From the plotted graph we can say that the data doesn't have a constant average as there are many leaps and troughs and also the variance is also different at different stages of the data. So, our data is not stationary.

Augmented Dickey Fuller Test

The Augmented Dickey-Fuller test is a type of statistical test called a unit root test. The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend. There are a number of unit root tests and the Augmented Dickey-Fuller may be one of the more widely used. It uses an auto-regressive model and optimizes an information criterion across multiple different lag values.

The null hypothesis of the test is that the time series can be represented by a unit root, that it is not stationary (has some time-dependent structure). The alternate hypothesis (rejecting the null hypothesis) is that the time series is stationary.

- Null Hypothesis (H0): If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.
- Alternate Hypothesis (H1): The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have time-dependent structure.

We interpret this result using the p-value from the test. A p-value below a threshold (such as 5% or 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationary).

- p-value > 0.05: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
- p-value <= 0.05: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

```
ADF Test Statistic : -2.287306605903843
p-value : 0.176094530447718
#Lags Used : 0
Number of Observations Used : 5096
weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary
```

Figure 14: The ADF test shows that our data is non-stationary

Stationarity Conversion with shift ()

Now let's convert our non-stationary data to stationary with shift () method. Here we take a shift () of 1 day which means all the records will step down to one step and we take the difference from the original data. Since we see a trend in our data, when we subtract today's value from yesterday's value considering a trend it will leave a constant value on its way thus making the plot stationary.

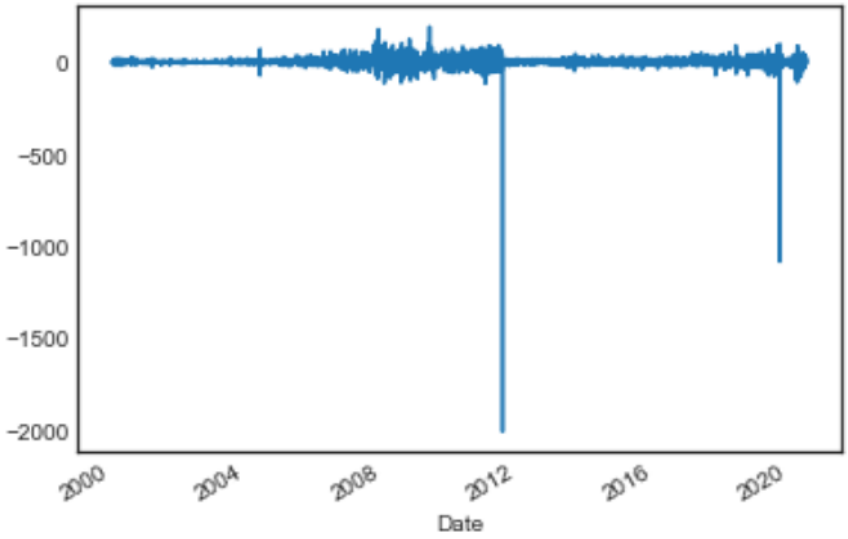

Figure 15: Stationarity Conversion with shift ()

Model Building – Forecasting & Prediction

Now we arrive at the most important phase of the project, which is essentially the objective of taking it up in the first place.

For model building, we are considering the close price feature as it is very reliable for prediction. VWAP is a derived/calculated value which doesn't make much sense while getting forecast value.

AUTO ARIMA – Autoregressive Integrated Moving Average

ARIMA stands for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

Any 'non-seasonal' time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.

An ARIMA model is characterized by 3 terms: p, d, q

where,
p is the order of the AR term
q is the order of the MA term
d is the number of differencing required to make the time series stationary

If a time series, has seasonal patterns, then you need to add seasonal terms and it becomes SARIMA, short for 'Seasonal ARIMA'.

Building AUTO ARIMA Model

For building the AUTO ARIMA model, we first split our training and testing data. For time series analysis we step back from using train test split as our data involves time and splitting can cause the mix of date across both train and test dataset which is vulnerable for a data leakage. So, we split based on the dates. Here, we split the training and validation data based on the year 2019 where the data before 2019 is training data and data after 2019 is validation data.

We train our model with AUTO ARIMA. Here the model selects the parameter p, q, and d value in a normal ARIMA model by itself by determining AIC value (Akaike Information Criterion). AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. AUTO ARIMA prefers the parameters which can reap less information loss.

```
Performing stepwise search to minimize aic
 ARIMA(2,0,2)(0,0,0)[0] intercept   : AIC=48100.711, Time=26.54 sec
 ARIMA(0,0,0)(0,0,0)[0] intercept   : AIC=48598.238, Time=7.90 sec
 ARIMA(1,0,0)(0,0,0)[0] intercept   : AIC=48164.905, Time=18.05 sec
 ARIMA(0,0,1)(0,0,0)[0] intercept   : AIC=48109.759, Time=26.44 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=83882.569, Time=6.56 sec
 ARIMA(1,0,2)(0,0,0)[0] intercept   : AIC=48115.424, Time=27.96 sec
 ARIMA(2,0,1)(0,0,0)[0] intercept   : AIC=48102.361, Time=24.44 sec
 ARIMA(3,0,2)(0,0,0)[0] intercept   : AIC=48099.789, Time=28.85 sec
 ARIMA(3,0,1)(0,0,0)[0] intercept   : AIC=48106.553, Time=31.20 sec
 ARIMA(4,0,2)(0,0,0)[0] intercept   : AIC=48101.286, Time=33.51 sec
 ARIMA(3,0,3)(0,0,0)[0] intercept   : AIC=48100.740, Time=30.15 sec
 ARIMA(2,0,3)(0,0,0)[0] intercept   : AIC=48098.969, Time=28.65 sec
 ARIMA(1,0,3)(0,0,0)[0] intercept   : AIC=48101.190, Time=30.30 sec
 ARIMA(2,0,4)(0,0,0)[0] intercept   : AIC=48100.400, Time=34.79 sec
 ARIMA(1,0,4)(0,0,0)[0] intercept   : AIC=48099.023, Time=31.51 sec
 ARIMA(3,0,4)(0,0,0)[0] intercept   : AIC=48102.314, Time=32.51 sec
 ARIMA(2,0,3)(0,0,0)[0]             : AIC=48096.853, Time=30.51 sec
 ARIMA(1,0,3)(0,0,0)[0]             : AIC=48099.029, Time=24.58 sec
 ARIMA(2,0,2)(0,0,0)[0]             : AIC=48098.538, Time=23.24 sec
 ARIMA(3,0,3)(0,0,0)[0]             : AIC=48098.684, Time=26.29 sec
 ARIMA(2,0,4)(0,0,0)[0]             : AIC=48098.344, Time=33.19 sec
 ARIMA(1,0,2)(0,0,0)[0]             : AIC=48113.394, Time=24.70 sec
 ARIMA(1,0,4)(0,0,0)[0]             : AIC=48096.963, Time=28.09 sec
 ARIMA(3,0,2)(0,0,0)[0]             : AIC=48097.746, Time=23.66 sec
 ARIMA(3,0,4)(0,0,0)[0]             : AIC=48100.265, Time=30.05 sec

Best model:  ARIMA(2,0,3)(0,0,0)[0]
Total fit time: 664.419 seconds
```

Figure 16: AUTO ARIMA prefers the parameters which can reap less information loss

From the AUTO ARIMA results, we got the values of p and q as 2 and 3 respectively and the AIC score is 48096.853.

Now we shall plot the forecasted values with the actual data.

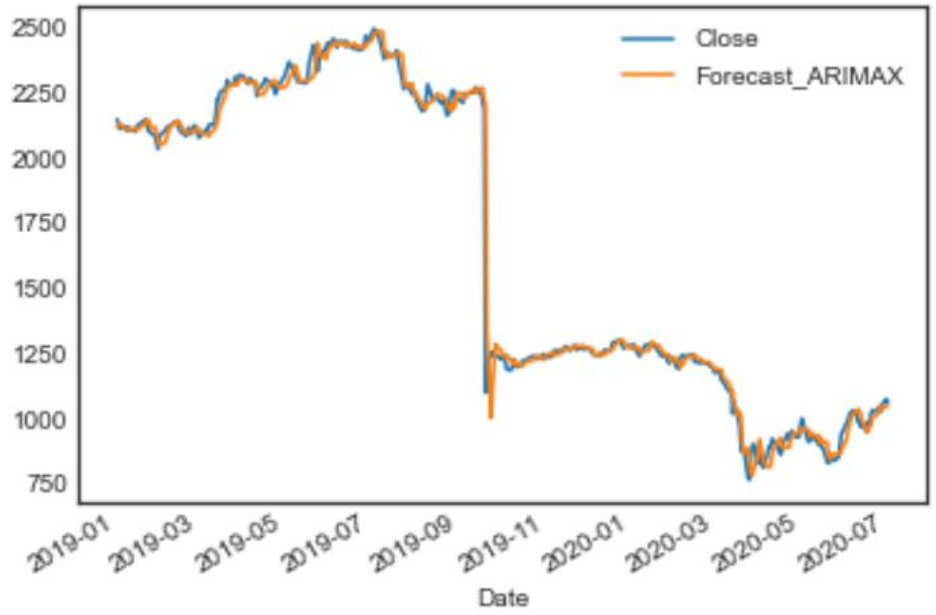

Figure 17: Our model has captured a good amount of information from the training dataset

Given how close the two lines are in the above figure, we can say that our model has captured a good amount of information from training dataset.

Let us now have a look at the performance metrics.

```
RMSE of Auto ARIMAX: 67.82958043595664

MAE of Auto ARIMAX: 26.95611741001754
```

Figure 18: Performance metrics of our model

We got the RMSE and MAE score of 67 and 26, which is a good score considering a time series data.

FB Prophet

Facebook developed an open sourcing Prophet, a forecasting tool available in both Python and R. It provides intuitive parameters which are easy to tune. Even someone who lacks deep expertise in time-series forecasting models can use this to generate meaningful predictions for a variety of problems in business scenarios.

We have created the future dataframe for 365 days. Since our dataset is till July 2020 only, we have dates till 2021 and now we are going to predict the stock prices for that.

Forecast Plot

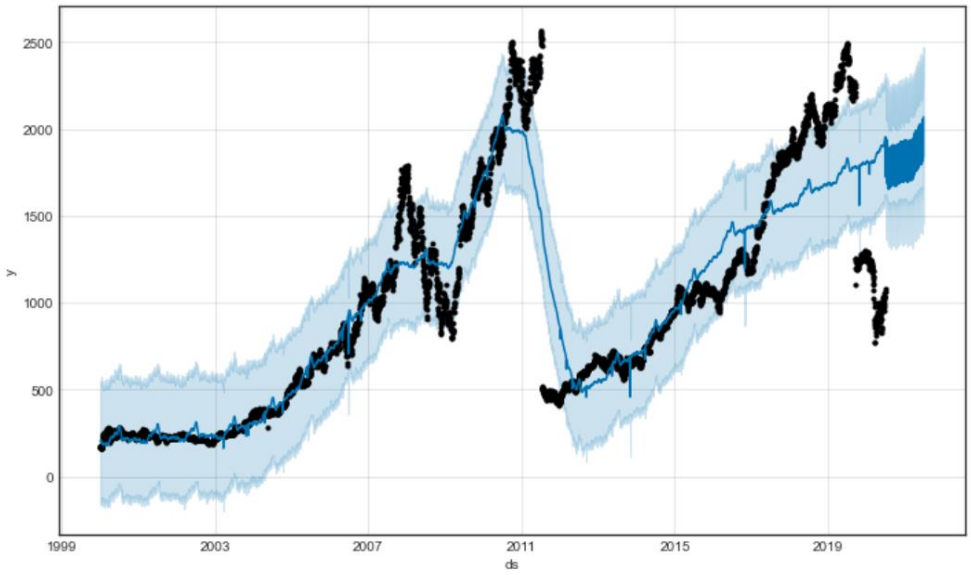

Figure 19: Forecast plot

From the plot we can see the predicted values in blue line, which follows most the actual trend. After 2020, we can see the blue line is extending upwards for the year 2021, which are the future prices in our model and we can say that the upward trend will continue in 2021. From hindsight, we can say our forecast is fairly accurate as the stock prices of HDFC Bank did indeed go up this year.

Forecast Components

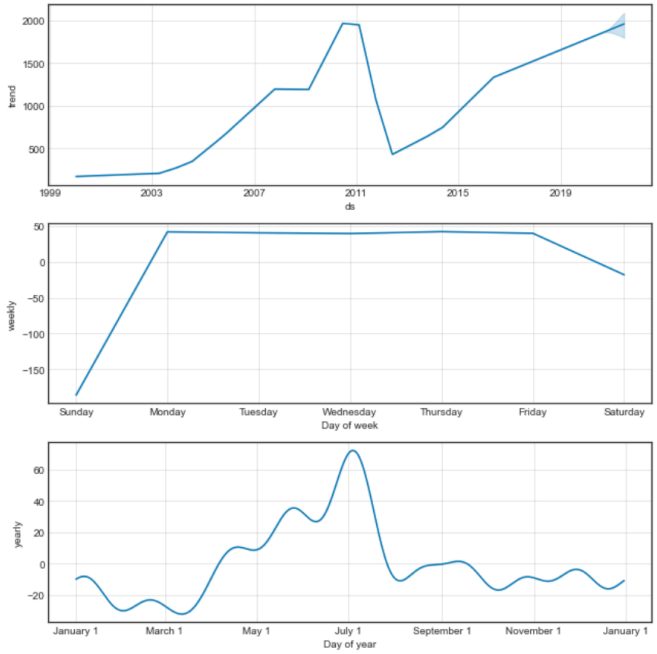

Figure 20: Forecast Components

From the model components of prophet we get the trend, weekly and yearly plots.

Conclusion

From our time series analysis of HDFC Bank, we saw how the company has evolved over the years to establish itself as the country’s largest private sector bank with annual revenue touching Rs. 1.21 lakh crores in 2021.