Final Report

Comparison of Various Time series models

Batch details	DSE November 2021
Team members	1. Dhanvanthri 2. Boobesh P 3. Prasanna Pprakash T S
Domain of Project	Finance & Risk Analytics
Proposed project title	Comparison of various Timeseries models
Group Number	6
Team Leader	Prasanna Pprakash T S
Mentor Name	Mr Mayuk Ghosh Mr Vikash Chandra

Industry Review

- Time series Forecasting has gone through several improvements in its long history
- The application and importance of it is only raising owing to the massive production of such data through technologies like internet of things and digitalisation of various industries and practices
- Time Series Forecasting is a technique of predicting Future happenings with based on past trends and assuming that future trends will be like past trends
- The application of Time series forecasting is wide ranging from weather forecasting to price forecasting
- Some Examples: Prediction of demand and supply, Prediction of rain, Prediction of power demand, Prediction of equipment failures and maintenance and so on

References

1. https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/
2. https://towardsdatascience.com/predicting-the-stock-market-is-hard-creating-a-machine-learning-model-probably-wont-help-e449039c9fe3
3. https://www.sciencedirect.com/science/article/pii/S1877050916311619
4. https://www.nseindia.com/

Dataset and Domain

Data Dictionary

Attribute	Definition	Datatype
Date	Date at which the price was noted	Datetime
Open Price	Current day’s open price	float
Close Price	Current day’s close price	float
High Price	Current day’s high price	float
Low Price	Current day’s low price	float
Day	Weekday	categorical

Pre-Processing Data Analysis

Variable categorization

- Our Dataset is made up of 4 float variables, 1 categorical and 1 Datetime variable

Pre-Processing-Data

- We have done pre-processing of data in Excel that is removed the variables Open Price, High Price, Low Price and kept the features Date, Close Price and Day
- As the continuation of above step, we have also removed the anomalies that is Sunday and Saturday entries which are exceptional trading days due to the occasion of Muharat trading and Budget session
- Once the removal of anomalies is done, we have dropped the feature Day from our dataset and only the required features that is Date and Close Price is kept
- We have 31 Null values in the close Price columns which are imputed using the backward fill
- There are No Outliers present in the dataset

Project Justification

Project Statement

We will be predicting the prices of stocks using various ML methods and see which is the best model for the stock ie. HDFC Bank

Complexity Involved

Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

- Fundamental Analysis involves analysing the company’s future profitability based on its current business environment and financial performance.
- Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market

As you might have guessed, our focus will be on the technical analysis part

Project Outcome

As a conclusion to this project, we will have the best model for HDFC Bank Stock with the least RMSE score

Data Exploration (EDA) & Feature Engineering

- Time series data is data that is measured at equally spaced intervals
- Think of a sensor that takes measurements every minute

<u>A sensor that takes measurements at random times is not time series</u>

1. As it's a time series data all the rows are dependent on each other
2. There is no need of scaling as we are doing univariate time series
3. There are no outliers present in the data

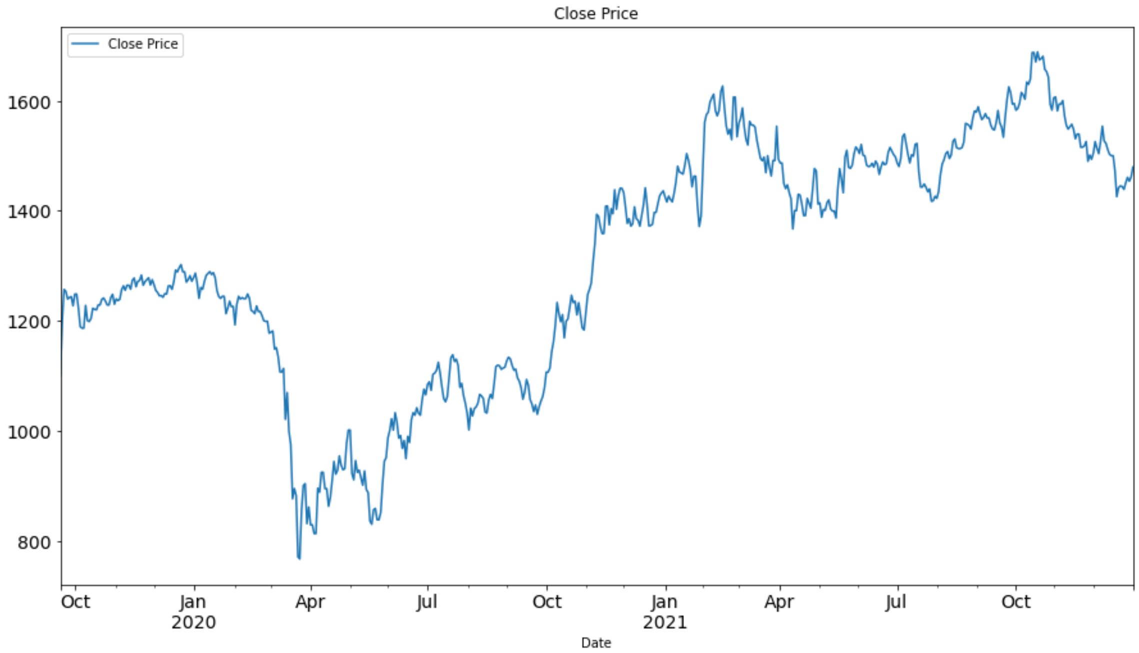


Figure 1 Line Plot of the HDFC bank for the Observed Timeline

Detection of Outliers – No Outliers Present

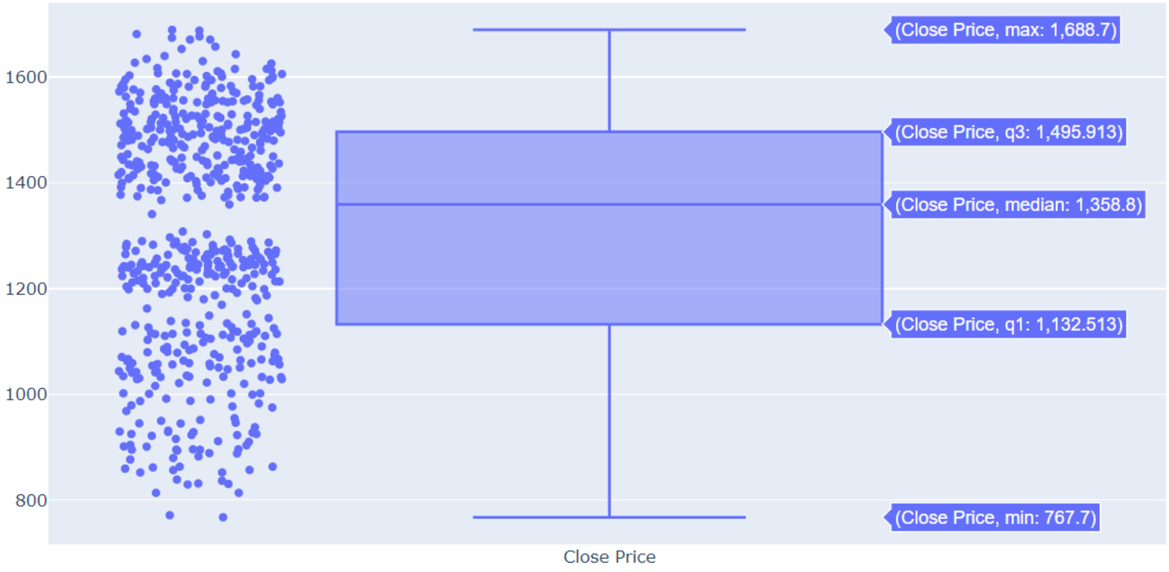


Figure 2 Box Plot

Missing Values Treatment

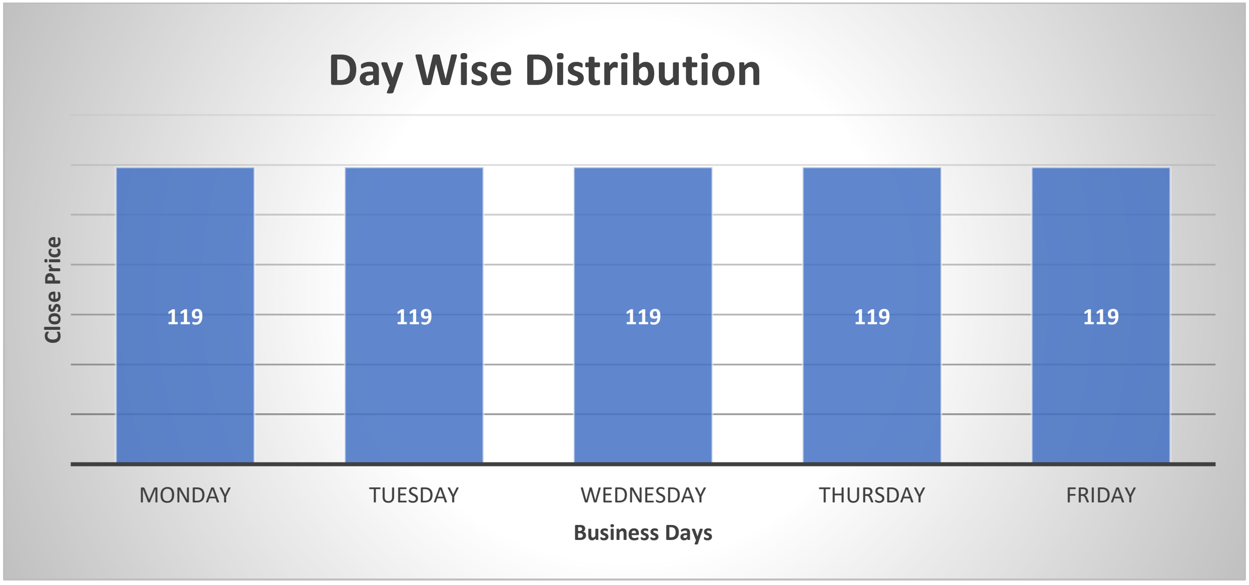
- There were 31 missing values
- We have used Backfill method to fill the missing values

Anomalies Treatment

- There were trading sessions happened on a Saturday and Sunday due to Budget sessions and Muhurat trading (Diwali holiday) which were dropped

Assumptions

1. As the time series need a data at regular intervals, we have taken range of data which gives equal distribution of Business Days



2. As the rows are not independent the data is sorted as per the dates

Decomposition

- Dealing with data that is sequential in nature requires special techniques. Unlike traditional Ordinary Least Squares or Decision Trees where the observations are independent, time series data is such that there is correlation between successive samples
- In other words, order very much matters

The first step on our journey is to identify the three components of time series data

1. Trend
2. Seasonality
3. Residuals

- Trend, as its name suggests, is the overall direction of the data
- Seasonality is a periodic component
- The residual is what’s left over when the trend and seasonality have been, You can think of them as a noise component

There are two major ways to decompose which are as follows

Additive

Additive is simply a sum of the three components [Additive = Trend + Seasonal + Residual]

$$O_t = T_t + S_t + R_t$$

O_t is the output

T_t is the trend

S_t is the seasonality

R_t is the residual

$_t$ is a variable representing a particular point in time

Multiplicative

It is a multiplication of all three components [Multiplicative = Trend * Seasonal * Residual]

$$O_t = T_t * S_t * R_t$$

Addictive vs Multiplicative

The primary question is how I can tell if a time series is additive or multiplicative

Simply plotting the original time series data is one way to do so

- If the seasonality and residual components are independent of the trend, then you have an **additive series**
- If the seasonality and residual components are in fact dependent, meaning they fluctuate on trend, then you have a **multiplicative series**

By looking at the plots below we can see that in additive decomposition the residuals components are dependent on trend hence we got ourselves a multiplicative series

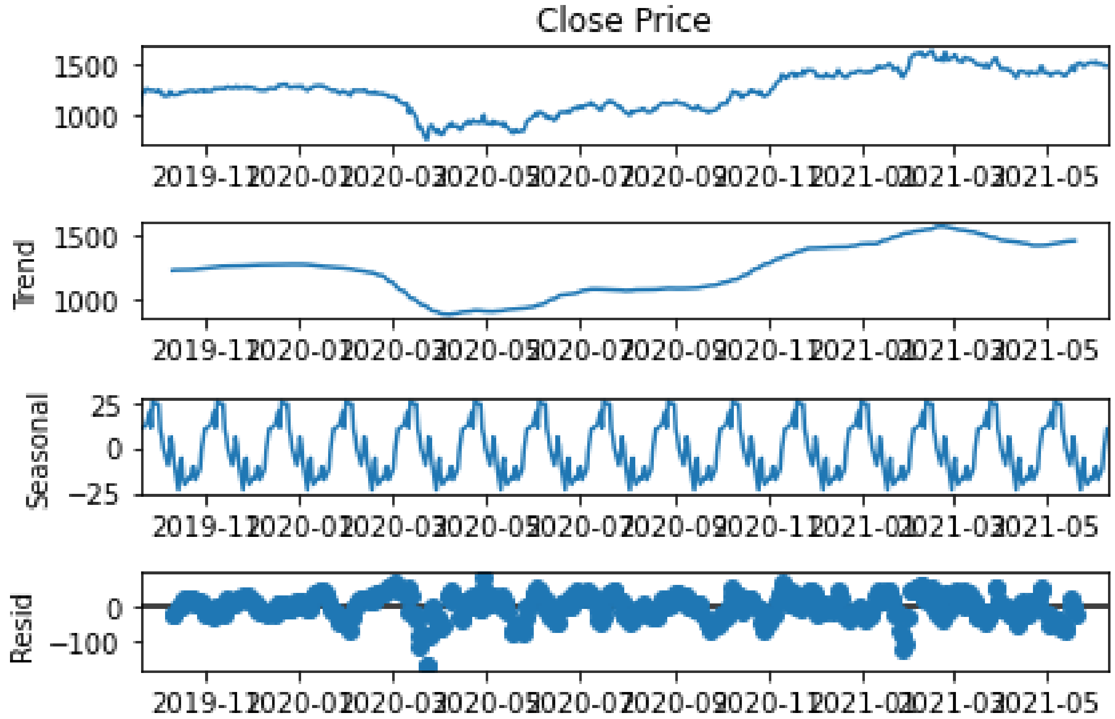


Figure 3 Additive Model

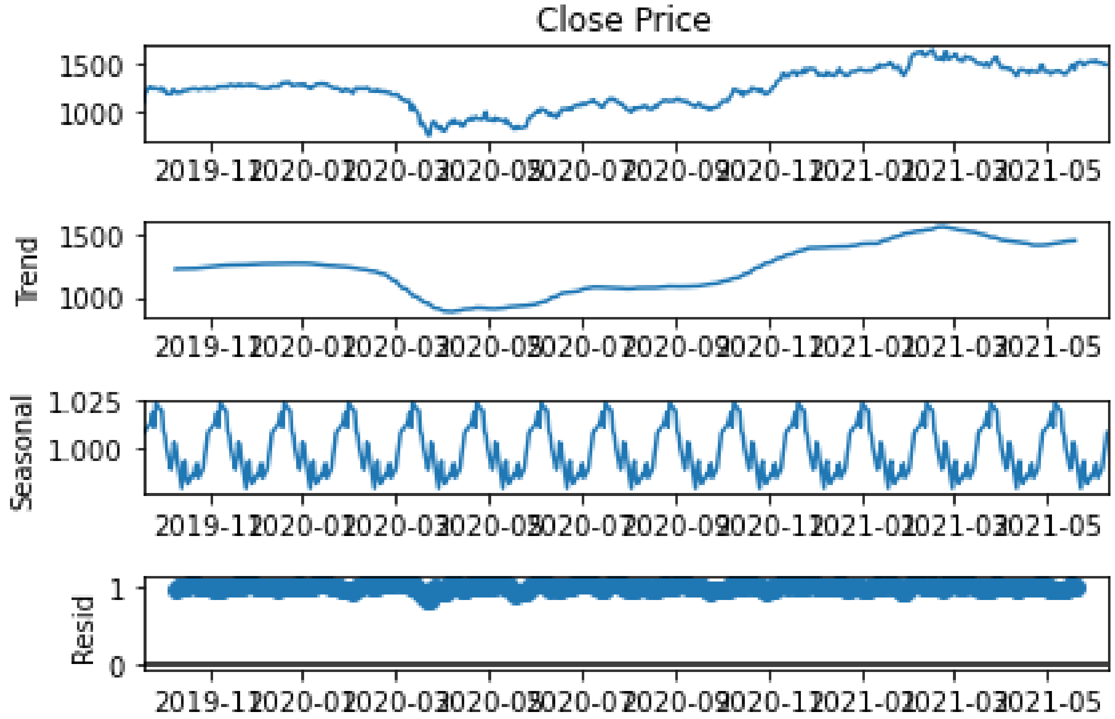


Figure 4 Multiplicative Model

Naïve Method

In Naïve Method forecasting the price for the next day, we simply take the last day value and estimate the same value for the next day. Such forecasting technique which assumes that the next expected point is equal to the last observed point is called **Naive Method**

$$\hat{y}_{t+1} = y_t.$$

Simple Average Method

In Simple Average forecasting technique, the expected value equal to the average of all previously observed points hence it's called **Simple Average technique**

$$\hat{y}_{x+1} = \frac{1}{x}\sum_{i=1}^{x} y_i$$

Moving Average Method

In a simple moving average model, we forecast the next value(s) in a time series based on the average of a fixed finite number 'p' of the previous values. Thus, for all i > p

$$\hat{y}_i = \frac{1}{p}(y_{i-1} + y_{i-2} + y_{i-3} \ldots\ldots + y_{i-p})$$

A moving average can be quite effective, especially if you pick the right p for the series and it's one of the most sort out method by the Traders

Simple Exponential Smoothing

In Simple exponential smoothing the forecasts are calculated using weighted averages where the weights decrease exponentially as observations come from further in the past, the smallest weights are associated with the oldest observations

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $0 \leq \alpha \leq 1$ is the **smoothing** parameter

The main shortcoming of the simple exponential model it doesn't take account of trend and seasonality of the data hence doesn't forecast for data with high variations

Holt's Linear Trend Method (Double Exponential Smoothing)

In Holt's Linear Trend Method, it considers the trend. It is nothing more than exponential smoothing applied to both level (the average value in the series) and trend.

$$\text{Forecast equation : } \hat{y}_{t+h|t} = \ell_t + h\, b_t$$

$$\text{Level equation : } \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1}+b_{t-1})$$

$$\text{Trend equation : } b_t = \beta*(\ell_t - \ell_{t-1})+(1-\beta)b_{t-1}$$

Again, the Holt's Linear Method doesn't take account of seasonality of the data

Holt's Winters Method (Triple Exponential Smoothing)

In Holt's Winters Method, it considers all. It is nothing more than exponential smoothing applied to all three level (the average value in the series), trend and season

$$\begin{array}{lrcl} \text{level} & L_t & = & \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{t-1}+b_{t-1}); \\ \text{trend} & b_t & = & \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}, \\ \text{seasonal} & S_t & = & \gamma(y_t - L_t) + (1-\gamma)S_{t-s} \\ \text{forecast} & F_{t+k} & = & L_t + kb_t + S_{t+k-s}, \end{array}$$

where s is the length of the seasonal cycle, for $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$

Stationarity of a Time Series

A Time Series is said to be stationary if its statistical properties such as mean, and variance remain constant over time

Most of the Time Series models work on the assumption that the time series data used to build a model is stationary

On other words, we can say that if a Time series has behaviour over time, then there is a high probability that it will follow the same in future

We can check the stationary of the data using Dickey-Fuller test whose hypothesis are as follows

Null Hypothesis– A unit root is present in a time series sample [Time Series is Stationary]

Alternate Hypothesis – A unit root is not present in a time series [Time Series is non-Stationary]

Making a Non-Stationary Time Series into Stationary Time Series

There are two major reasons behind a non-stationary time series

- Trend – varying mean over time
- Seasonality – variations at specific timeframes

Again, there are two ways to remove trend and seasonality

- Differencing
- Decomposition

Differencing

One of the most common methods of dealing with both trend and seasonality is differencing

In this we take the difference of the observation at a particular instant with that of the previous instant

Decomposing

In this method both trend and seasonality are modelled separately, and the remaining part of the series is returned

ARIMA

It stands for Autoregressive Integrated Moving Average. While exponential smoothing models were based on a description of trend and seasonality in the data, ARIMA model aim to describe the correlations in the data with each other

It works on the assumption that the time series data which is passed into the model is stationary

The predictors depend on the parameters (p, d, q) of the ARIMA model:

1. **Number of AR (Auto-Regressive) terms (p):** AR terms are just lags of dependent variable. For instance, if p is 5, the predictors for x(t) will be x(t-1) ….x(t-5).
2. **Number of MA (Moving Average) terms (q):** MA terms are lagged forecast errors in prediction equation. For instance, if q is 5, the predictors for x(t) will be e(t-1) ….

e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.

3. **Number of Differences (d):** These are the number of nonseasonal differences, i.e., in this case we took the first order difference. So, either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.

An importance concern here is how to determine the value of ‘p’ and ‘q’. We use two plots to determine these numbers. Let’s discuss them first.

1. **Autocorrelation Function (ACF):** It is a measure of the correlation between the TS with a lagged version of itself. For instance, at lag 5, ACF would compare series at time instant ‘t1’…’t2’ with series at instant ‘t1-5’…’t2-5’ (t1-5 and t2 being end points).
2. **Partial Autocorrelation Function (PACF):** This measures the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. E.g., at lag 5, it will check the correlation but remove the effects already explained by lags 1 to 4.

- *ARIMA, is one of the most widely used forecasting methods for univariate time series data forecasting*
- *Although the method can handle data with a trend, it does not support time series with a seasonal component*

SARIMA

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component

It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality

Configuring a SARIMA requires selecting hyperparameters for both the trend and seasonal elements of the series

Trend Elements

There are three trend elements that require configuration.

They are the same as the ARIMA model; specifically:

- **p**: Trend autoregression order
- **d**: Trend difference order
- **q**: Trend moving average order

Seasonal Elements

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

- **P**: Seasonal autoregressive order
- **D**: Seasonal difference order
- **Q**: Seasonal moving average order
- **S**: The number of time steps for a single seasonal period

In Seasonal ARIMA model, seasonal AR and MA terms predict (x{t}) using data values and errors at times with lags that are multiples of S (the span of the seasonality)

- A Seasonal first order MA (1) model (with S=12) would use (x{t-12}) as a predictor and a seasonal second order MA (2) model would use (x{t-12}) and (x{t-24})

Seasonal Differencing

It is defined as a difference between a value and a value with lag that is multiple of S

- With S=12, which may occur with monthly data, seasonal difference would be like x{t} = x{t}-x{t-12}

Seasonal differencing removes seasonal trend

FB Prophet

The Prophet library is an open-source library designed for making forecasts for univariate time series datasets

It is easy to use and designed to automatically find a good set of hyperparameters for the model to make skilful forecasts for data with trends and seasonal structure by default

Prophet is robust to missing data and shifts in the trend, and typically handles outliers well

Prophet implements what they refer to as an additive time series forecasting model, and the implementation supports trends, seasonality, and holidays

Fitting the Model

To use Prophet for forecasting, first, a Prophet () object is defined and configured, then it is fit on the dataset by calling the fit () function and passing the data

The Prophet () object takes arguments to configure the type of model you want, such as the type of growth, the type of seasonality, and more. By default, the model will work hard to figure out almost everything automatically

The fit () function takes a Data Frame of time series data. The Data Frame must have a specific format. The first column must have the name '*ds*' and contain the date-times. The second column must have the name '*y*' and contain the observations

Deployment

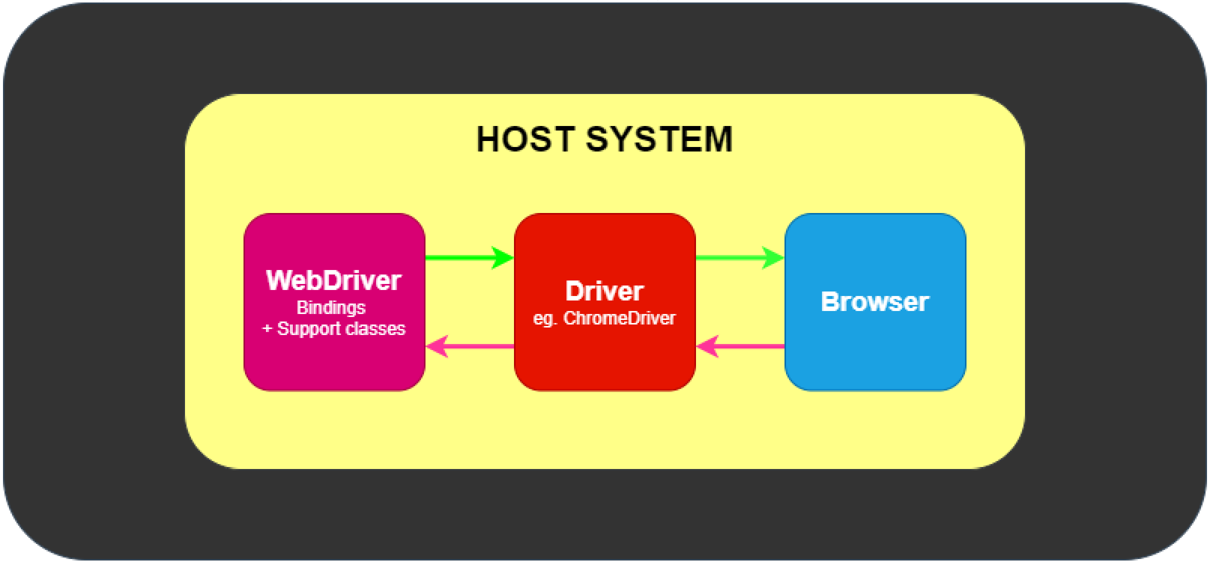
Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data

As the last stages of our Time Series project cycle, we have written a program to automate the buy and selling of stocks in the Trading View website using Selenium

The following libraries are used:

Selenium

- It is an umbrella project for a range of tools and libraries that enable and support the automation of web browser
- WebDriver uses browser automation APIs provided by browser vendors to control browser and run tests
- This is as if a real user is operating the browser
- Since WebDriver does not require its API to be compiled with application code, it is not intrusive
- Hence, you are testing the same application which you push live



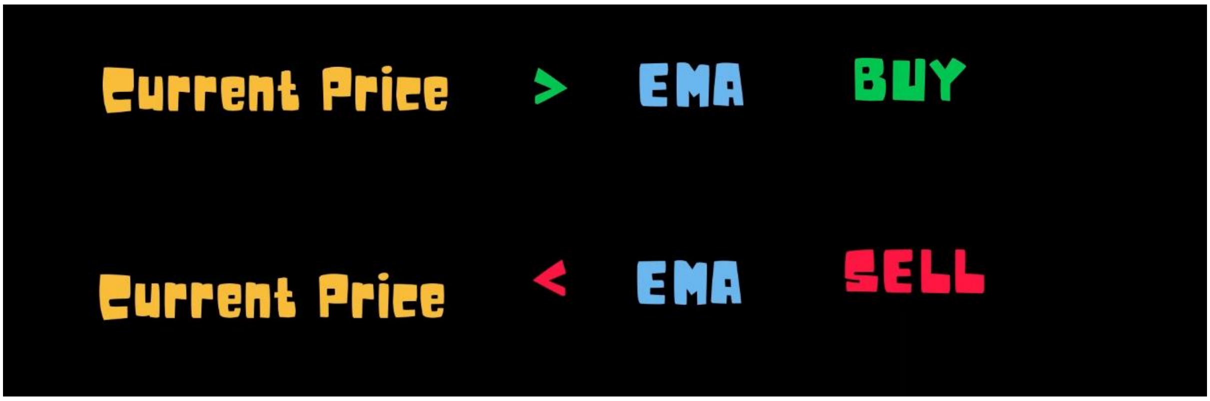
tradingview_ta

- TradingView_TA is an unofficial Python API wrapper to retrieve technical analysis from TradingView
- It is used to get technical analysis of stock

We have used couple of simple indicators which are widely used by Intra Day traders to do their trades

Exponential Moving Average

- Exponential moving averages (EMA) is a weighted average that gives greater importance to the price of a stock in more recent days, making it an indicator that is more responsive to new information



Current Price > EMA BUY

Current Price < EMA SELL

Relative Strength Index

- The relative strength index (RSI) is most used to indicate temporarily overbought or oversold conditions in a market
- The RSI measures the power behind price movements over a recent period, typically 14 days
- Recommended strategy using the RSI is by using the **30 and 70 levels in** the oscillator as **oversold and overbought levels** respectively
- This means that when RSI falls below 30, you aim to buy the financial security that has been sold too much and when the RSI reaches over 70, you aim to sell the financial asset that has been bought too much