# Stock Price Movement and Prediction (NIFTY-50 Stocks)

## Business Problem:

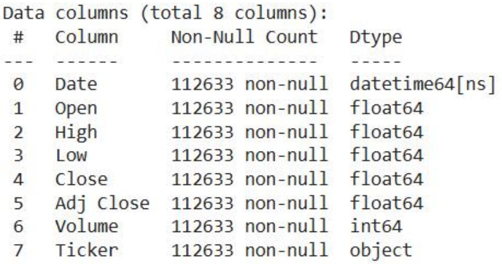
Customers are interested in understanding what stocks to invest in, given data about a stock and its past performance. This would help the customers make an educated investment choice. Therefore, given a particular stock and its information, the goal is to be able to predict or forecast how the stock would do in the future.

## Data Dictionary (Yahoo Finance Data):

**About the Dataset -** The **NIFTY-50** data set comprises historical stock market data for the 50 most significant and actively traded stocks listed on the **National Stock Exchange of India (NSE)**. These stocks represent a diverse array of sectors, including banking, technology, energy, pharmaceuticals, and consumer goods, providing a comprehensive overview of the Indian economy**.** Below is the data-dictionary -

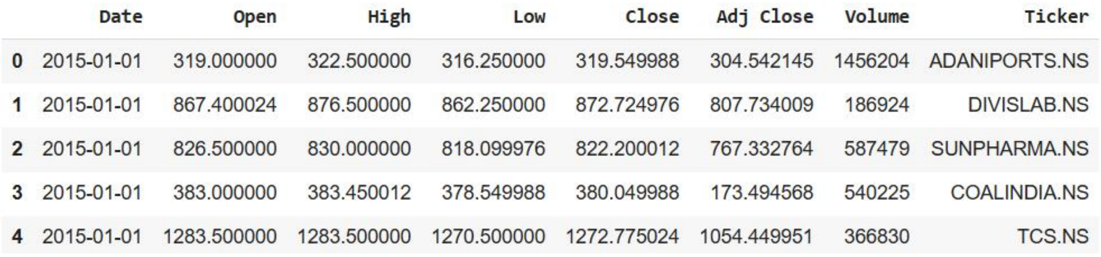
- **Date:** Represents the specific day the stock data was recorded
- **Open:** The first price at which the stock was traded when the market opened. It can be used to gauge the market's initial reaction to news and events
- **High:** The maximum price at which the stock was traded during the session. It indicates the stock's peak performance on a given day
- **Low:** The minimum price at which the stock was traded during the session. It provides insight into the stock's lowest performance on a given day
- **Close:** The final price at which the stock was traded when the market closed. It is often used as a reference point for the stock's daily performance
- **Adj Close:** Adjusted Close takes into account all corporate actions and adjustments. This is especially useful for back-testing and historical analysis as it reflects the stock's actual value more accurately
- **Volume:** Indicates the total number of shares traded during the day. High volume can indicate strong interest in the stock, while low volume can suggest weak interest
- **Ticker:** A unique identifier for the stock, typically an abbreviation of the company name or another identifier used on the stock exchange

**Data Type overview -**



```
Data columns (total 8 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   Date       112633 non-null  datetime64[ns]
 1   Open       112633 non-null  float64
 2   High       112633 non-null  float64
 3   Low        112633 non-null  float64
 4   Close      112633 non-null  float64
 5   Adj Close  112633 non-null  float64
 6   Volume     112633 non-null  int64
 7   Ticker     112633 non-null  object
```

**Quick Glance at the Dataset -**

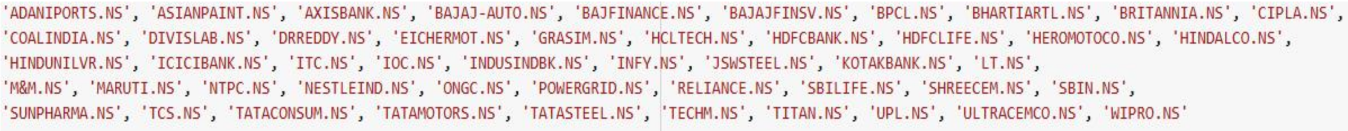


| | Date | Open | High | Low | Close | Adj Close | Volume | Ticker |
|---|---|---|---|---|---|---|---|---|
| **0** | 2015-01-01 | 319.000000 | 322.500000 | 316.250000 | 319.549988 | 304.542145 | 1456204 | ADANIPORTS.NS |
| **1** | 2015-01-01 | 867.400024 | 876.500000 | 862.250000 | 872.724976 | 807.734009 | 186924 | DIVISLAB.NS |
| **2** | 2015-01-01 | 826.500000 | 830.000000 | 818.099976 | 822.200012 | 767.332764 | 587479 | SUNPHARMA.NS |
| **3** | 2015-01-01 | 383.000000 | 383.450012 | 378.549988 | 380.049988 | 173.494568 | 540225 | COALINDIA.NS |
| **4** | 2015-01-01 | 1283.500000 | 1283.500000 | 1270.500000 | 1272.775024 | 1054.449951 | 366830 | TCS.NS |

## List of NIFTY-50 Stocks -

**Note** - Of the original 50 stocks, **‘HDFC.NS’** got delisted by Yahoo finance. Due to this, the dataset contains only 49 unique stocks.

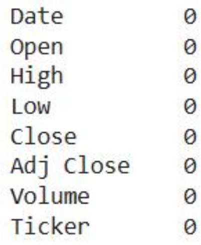
Below is a list of the 49 stocks included in the dataset -



```
'ADANIPORTS.NS', 'ASIANPAINT.NS', 'AXISBANK.NS', 'BAJAJ-AUTO.NS', 'BAJFINANCE.NS', 'BAJAJFINSV.NS', 'BPCL.NS', 'BHARTIARTL.NS', 'BRITANNIA.NS', 'CIPLA.NS',
'COALINDIA.NS', 'DIVISLAB.NS', 'DRREDDY.NS', 'EICHERMOT.NS', 'GRASIM.NS', 'HCLTECH.NS', 'HDFCBANK.NS', 'HDFCLIFE.NS', 'HEROMOTOCO.NS', 'HINDALCO.NS',
'HINDUNILVR.NS', 'ICICIBANK.NS', 'ITC.NS', 'IOC.NS', 'INDUSINDBK.NS', 'INFY.NS', 'JSWSTEEL.NS', 'KOTAKBANK.NS', 'LT.NS',
'M&M.NS', 'MARUTI.NS', 'NTPC.NS', 'NESTLEIND.NS', 'ONGC.NS', 'POWERGRID.NS', 'RELIANCE.NS', 'SBILIFE.NS', 'SHREECEM.NS', 'SBIN.NS',
'SUNPHARMA.NS', 'TCS.NS', 'TATACONSUM.NS', 'TATAMOTORS.NS', 'TATASTEEL.NS', 'TECHM.NS', 'TITAN.NS', 'UPL.NS', 'ULTRACEMCO.NS', 'WIPRO.NS'
```

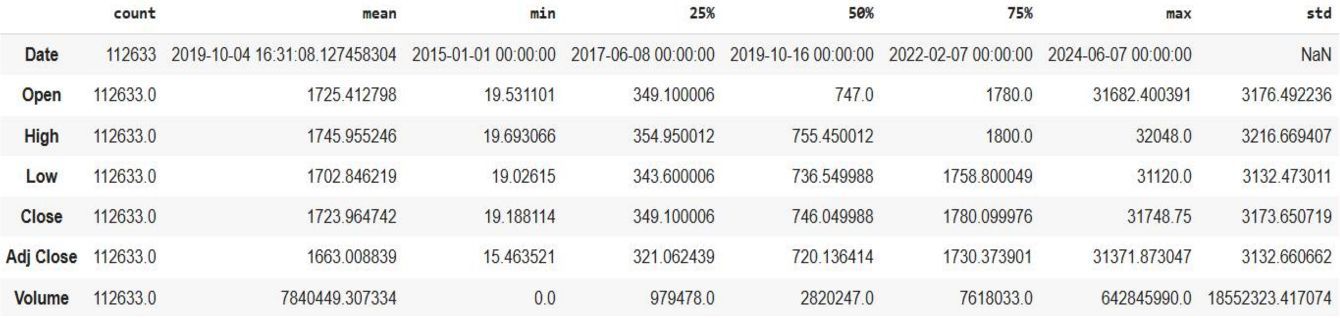
### Missing Data -

There is no missing data coming from the Yahoo Finance API as seen below -



```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
Ticker       0
```

### Summary Statistics -



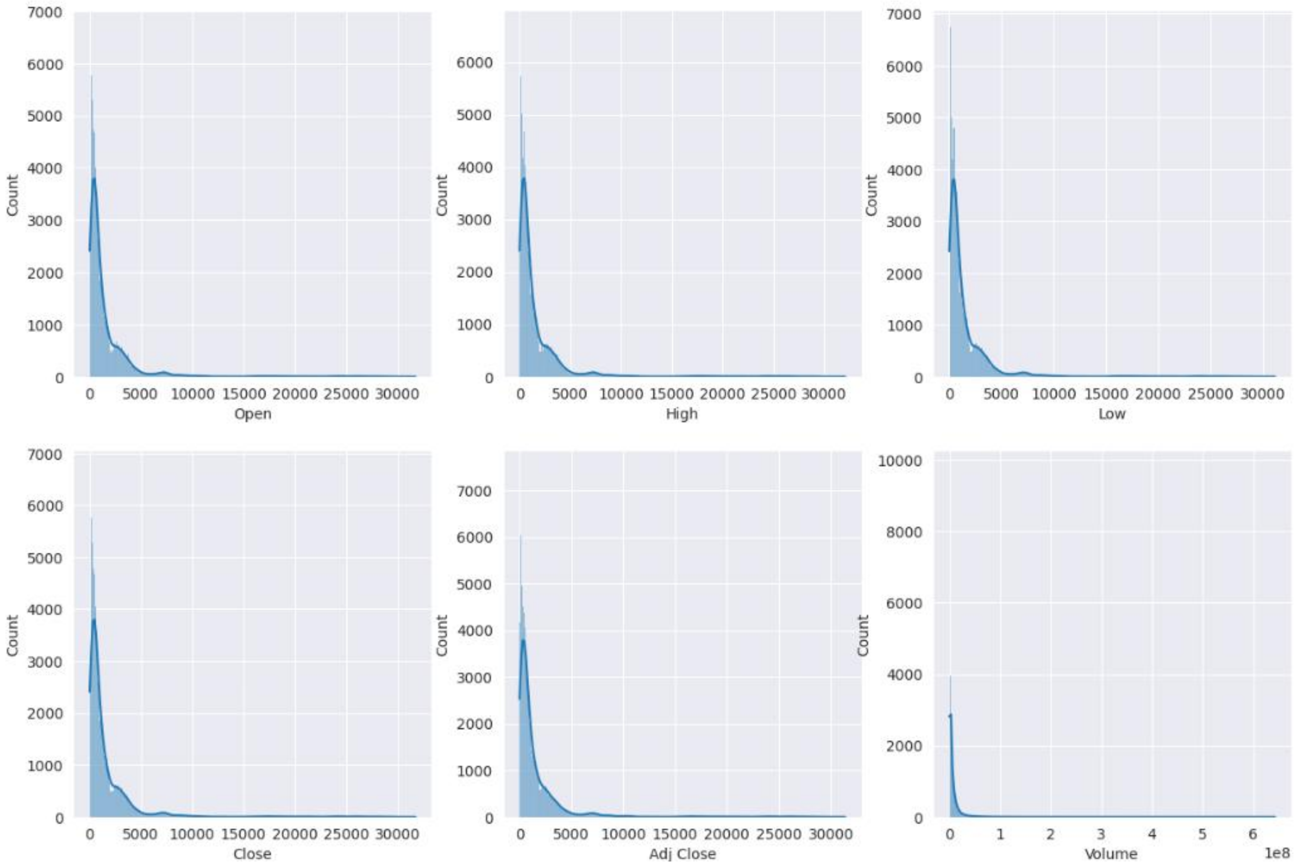
| | count | mean | min | 25% | 50% | 75% | max | std |
|---|---|---|---|---|---|---|---|---|
| **Date** | 112633 | 2019-10-04 16:31:08.127458304 | 2015-01-01 00:00:00 | 2017-06-08 00:00:00 | 2019-10-16 00:00:00 | 2022-02-07 00:00:00 | 2024-06-07 00:00:00 | NaN |
| **Open** | 112633.0 | 1725.412798 | 19.531101 | 349.100006 | 747.0 | 1780.0 | 31682.400391 | 3176.492236 |
| **High** | 112633.0 | 1745.955246 | 19.693066 | 354.950012 | 755.450012 | 1800.0 | 32048.0 | 3216.669407 |
| **Low** | 112633.0 | 1702.846219 | 19.02615 | 343.600006 | 736.549988 | 1758.800049 | 31120.0 | 3132.473011 |
| **Close** | 112633.0 | 1723.964742 | 19.188114 | 349.100006 | 746.049988 | 1780.099976 | 31748.75 | 3173.650719 |
| **Adj Close** | 112633.0 | 1663.008839 | 15.463521 | 321.062439 | 720.136414 | 1730.373901 | 31371.873047 | 3132.660662 |
| **Volume** | 112633.0 | 7840449.307334 | 0.0 | 979478.0 | 2820247.0 | 7618033.0 | 642845990.0 | 18552323.417074 |

### Key Observations about the Dataset -

1. There are a total of 112,633 rows in the dataset
2. Stock data for the NIFTY-50 stocks have been fetched going back until Jan 2015
3. Stock Market operates on the week-days and thus we do not have data for the weekends
4. The **"Adj Close"** column is what we would want to predict later as its more useful for historical analysis and back testing

# Exploratory Data Analysis

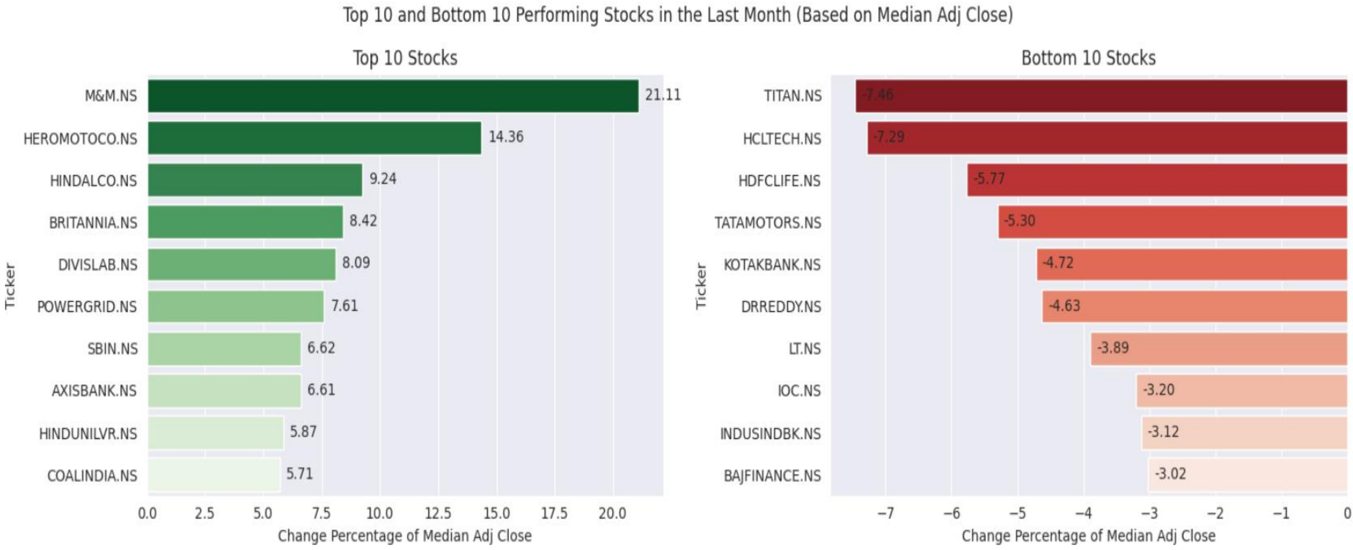
### Distribution of the Numerical Features -



### Observations -

- The features have a right-skewed distribution, indicating the presence of some higher-than-usual priced stocks that could also be premium stocks

### Analysis of Top/Bottom Performing Stocks based on Median Growth in the Last 1 Month -

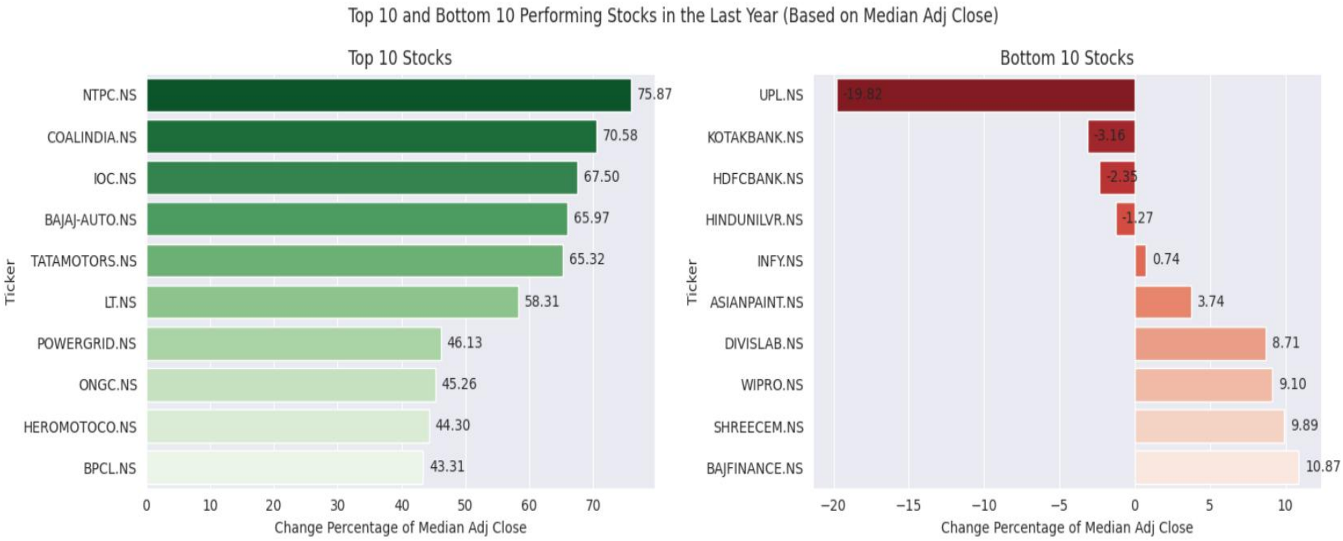


**Observations from the Monthly Analysis -**

- Based on the monthly analysis of stock adjusted closing prices, we observe that the following stocks have grown the most as compared to their last month median adjusted closing values. These could net good gains in the short term:
    1. M&M.NS
    2. HEROMOTOCO.NS
    3. HINDALCO.NS

- Similarly, the following stocks have performed the worst as compared to the last month median adjusted closing values:
    1. HCLTECH.NS
    2. TITAN.NS
    3. HDFCLIFE.NS

**Analysis of Top/Bottom Performing Stocks based on Median Growth in the Last 1 Year -**

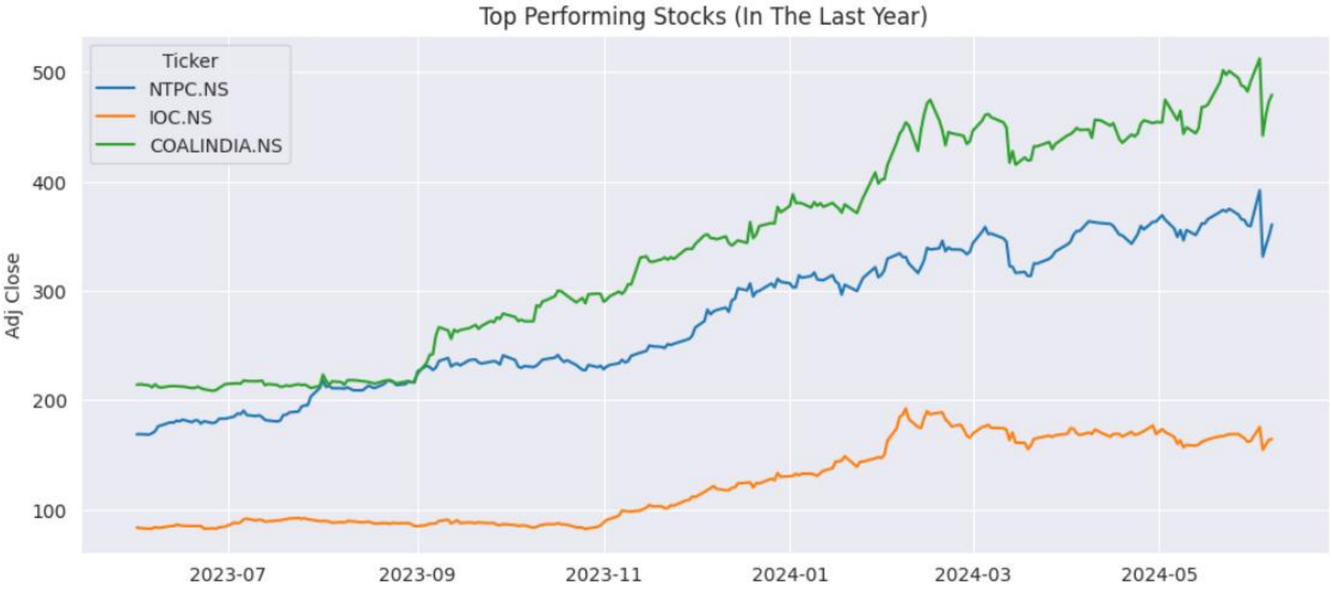


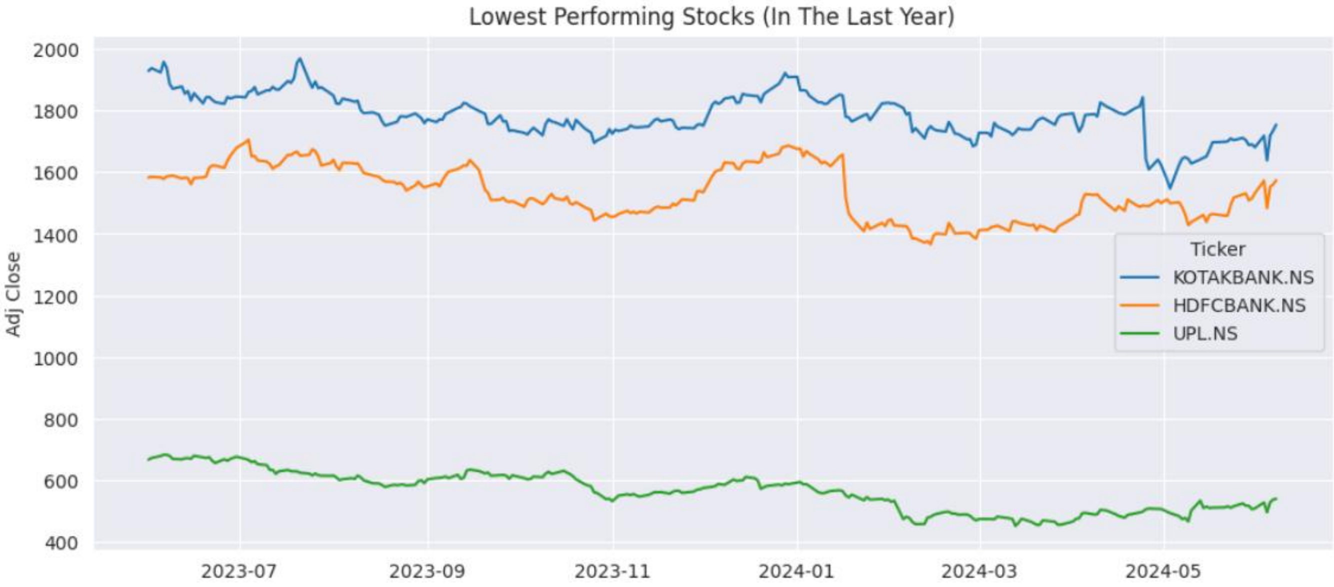
**Observations from the Yearly Analysis -**

- Based on the yearly analysis of stock adjusted closing prices, we observe that the following stocks have grown the most as compared to their last year median adjusted closing values. These could be seen as great long-term investment choices:
    1. **NTPC.NS**
    2. **COALINDIA.NS**
    3. **IOC.NS**

- Similarly, the following stocks have performed poorly as compared to their previous year median adjusted closing values:
    1. **UPL.NS**
    2. **KOTAKBANK.NS**
    3. **HDFCBANK.NS**

**Analysis of Top 3 and Bottom 3 Performers -**



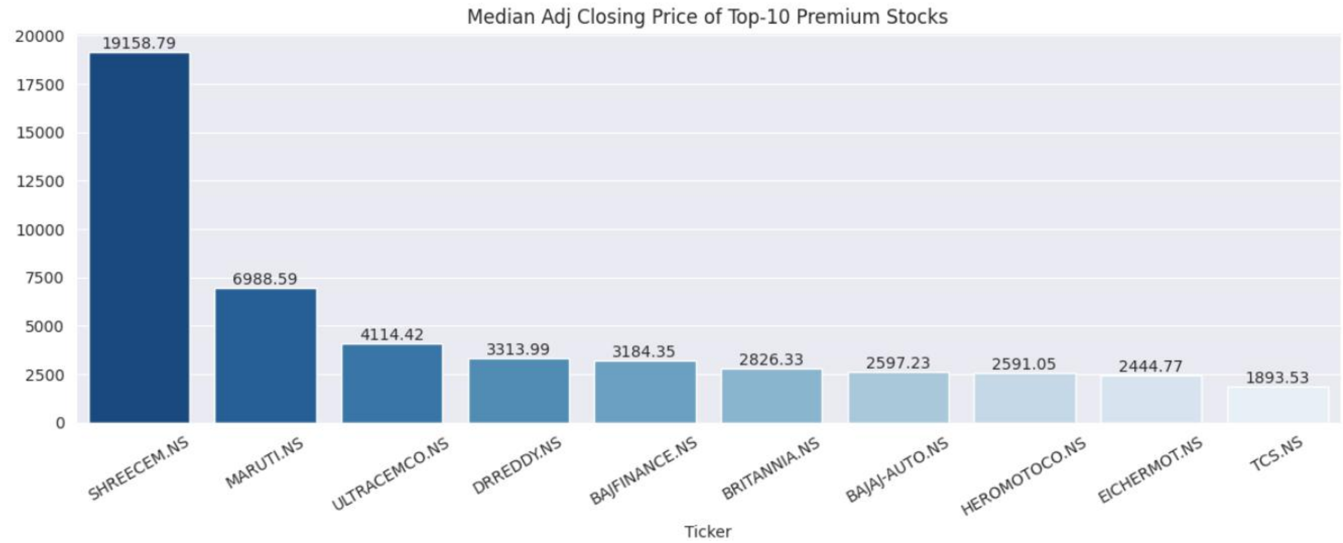


**Observations -**

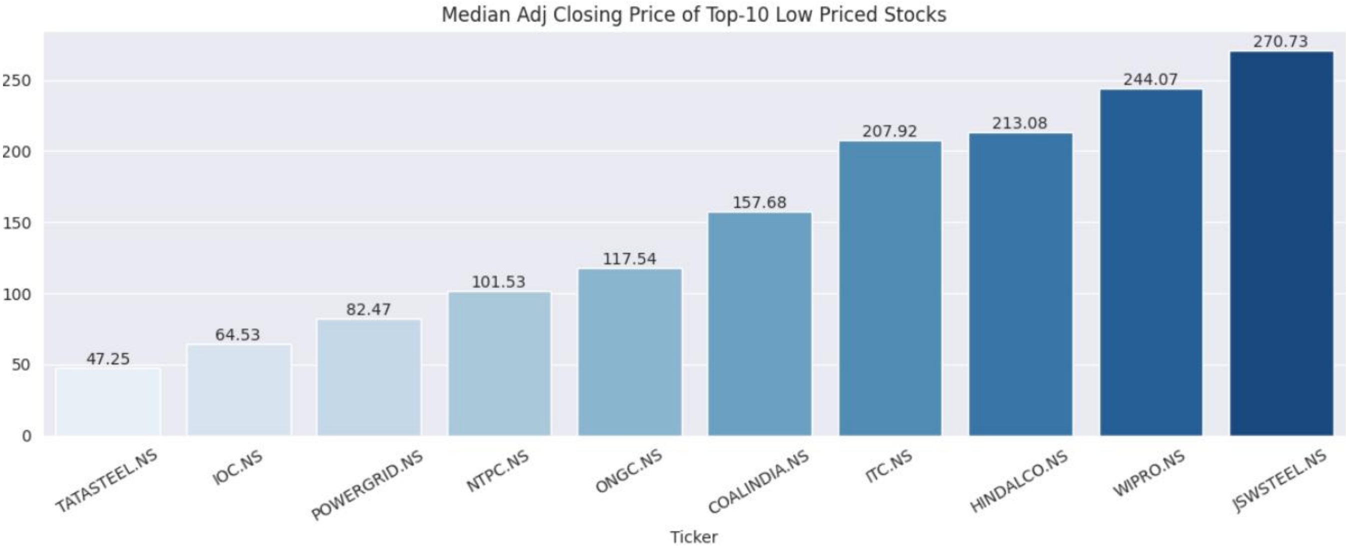
- The above charts highlight how the top performing stocks have a somewhat-consistent upward direction
- Similarly, the poor performers have a downward trend over the last year
- What's important to notice is that a stock's past performance can carry over important information to how the stock may perform today. This can be useful for crafting useful features for the model
- An interesting difference between the low and the high performing stocks is that the low performing stocks have a much-higher adj closing price value as compared to the adj closing price of the high-performing stocks

**Analyzing Stocks by their Adj Closing Price -**

**Top 10 Premium Stocks (Based on Median Adj Closing Price)**



**Top 10 Affordable Stocks (Based on Median Adj Closing Price)**



**Observations -**

- **SHREECEM.NS**, **MARUTI.NS** and **ULTRACEMCOS.NS** are the top 3 most-expensive stock options. These could be considered premium
- **TATASTEEL.NS**, **IOC.NS** and **POWERGRID.NS** are the top 3 least-expensive stock options available

# Modelling Approach and Set-up -

### Approach -

There are two ways to go about when it comes to predicting and forecasting stocks. They are:

1. **Predicting/Forecasting the Closing Price of a Stock** - This would mean that we predict the closing price for the stock tomorrow
2. **Predicting the Stock Price Direction** - This would mean that we predict whether a stock's closing price is going to increase or decrease tomorrow. This would break down the problem into a binary-classification problem

As the stock-market is very volatile and has extremely non-linear patterns that are heavily influenced by external factors such as politics, sentiment of a brand, etc, predicting the price of the stock for the next day is a near-impossible task. A slightly easier task would be to attempt to predict stock price down the line (say 3 months in advance). However, even that would be a very difficult task to solve.

Instead, if we're able to identify stocks that may increase or decrease in price for the next day, it could bring a lot of value to the customers, as they can then make informed and educated investments. This would thus mean that the task would be a binary classification task.

### Modelling Setup -

- **Positive Class** - Stock Price Increasing / Moving Up the next day
- **Negative Class** - Stock Price Decreasing / Moving Down the next day

With this model set-up we can predict what stocks have the highest probability to go up in price the next day.

The long running goal here is to suggest what stocks would go up in price the next day. Instead of simply predicting that a stock would go up or down, we can predict the stocks that are most likely to go up (probability)

### Metrics to use -

In this context of this binary classification problem, **False Positive** has a higher weightage as compared to a **False Negative** as we want to make sure that what the model predicts is correct. Thus precision would make a lot of sense here.

Here,

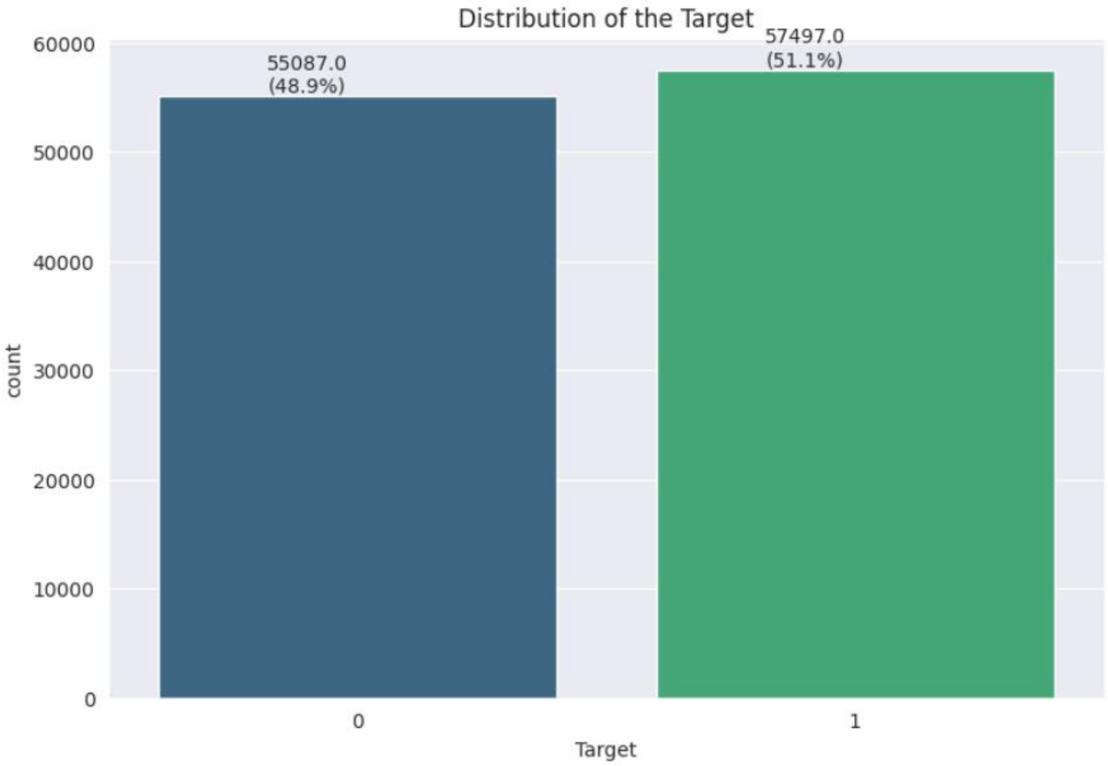
- **False Positive** - Incorrectly predicting that a stock’s price would go up tomorrow, when in reality it went down
- **False Negative -** Incorrectly predicting that a stock’s price would go down tomorrow, when in reality it went up

Based on the definition of False Positive and False Negative, we can tell that False Positives carry a larger risk. Thus,

- **Primary Metric:** Precision Score
- **Supporting Metric:** Confusion Matrix

## Creating the Target Column

The target can be created by checking whether tomorrow's adjusted closing price for a particular stock increases or decreases from today's adjusted closing price. If it increases, then assign Positive class, else Negative. Below is the class distribution post this step -

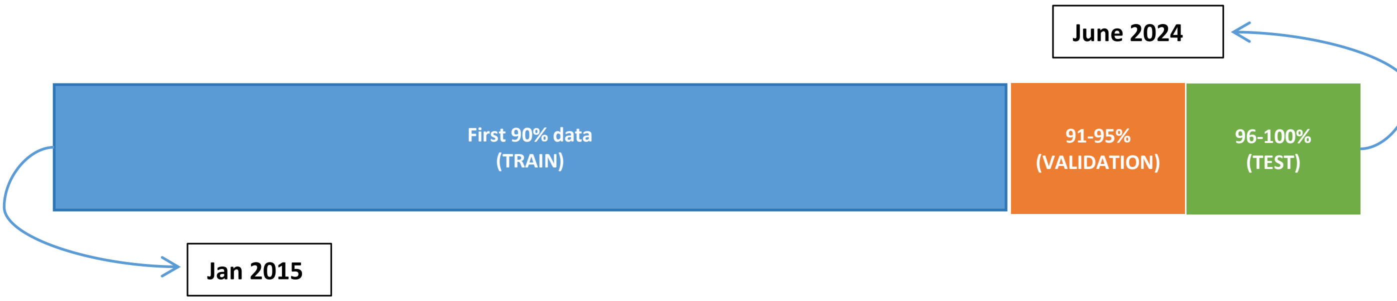


We observe that the targets are relatively balanced in the dataset.

## Train-Validation-Test Split

Time series data, unlike, other data encountered in ML, cannot be split randomly into training, testing and validation sets. As the data has a time-nature to it, it has to be split along the time axis.

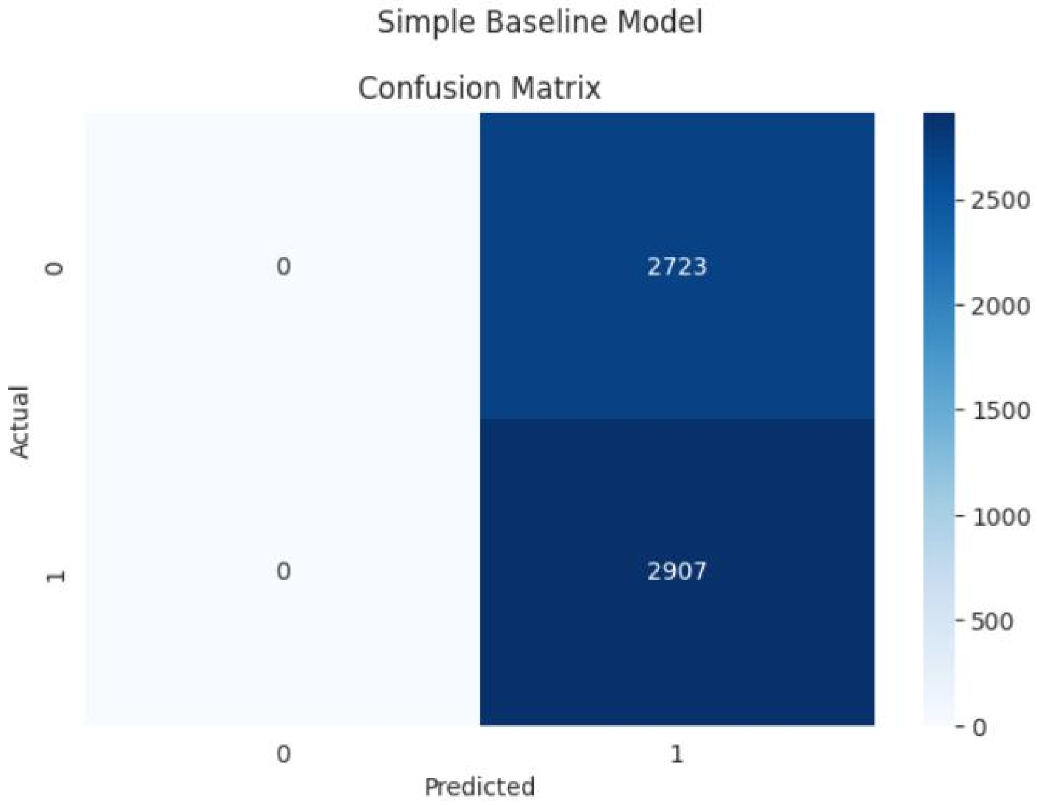
This is how the data has been split in the case study using Time-Based Splitting -



## Baseline Model -

In order to judge performance of models built later, a baseline model is required. Baseline model is the lowest guaranteed performance that we can achieve on the test set using heuristic driven methods or simple models. In this case, we would just predict every data point in the test set to belong to positive class.

```
Precision Score: 0.52
```



**Observations -**
The simple heuristic based baseline model achieves a Precision Score of **0.52**.

## Models experimented with -

Due to the extremely non-linear nature of the stock market, ensemble techniques would ideally work best. Models used are:

1. Random Forest
2. XGBoost
3. XGBoost + Feature Engineering
4. XGBoost + Feature Engineering + Hyper Parameter Tuning

## Feature Engineering -

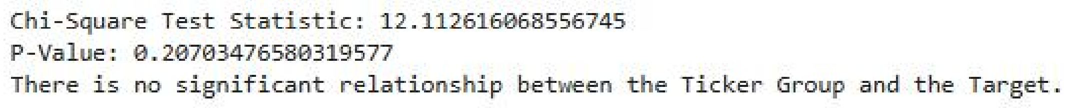
- **Stock direction** - Indicates whether the stock's closing price today is greater than yesterday's closing price
- **Stock Close Price Percentage Change from Yesterday** - Calculates the Change in Percentage in the stock's adjusted closing price as compared to yesterday adjusted closing price
- **Stock Price Fluctuation** - Difference between the Highest Price of the Stock and the Lowest Price for that day
- **Stock Price Difference** - Difference between the Closing Price and the Opening price of the stock for that day
- **Rolling averages (2 day, 5 day, 60 day, 250 day)** - Computes the 2 day, 5 day, 60 day and 250 day rolling averages of the Adjusted Closing Price for all stocks
- **Closing Ratios (2 day, 5 day, 60 day, 250 day)** - Ratio of Today's Adj Close Price / Rolling Avg (2 day, 5 day, 60 day and 250 day). This ratio can tell whether the market has been doing well or been down for some time
- **Horizon Trends (2 day, 5 day, 60 day, 250 day**) - Number of times stock price went up in the specified window. This is a rolling sum on the target in the specified window

## Other Features added -

**Stock Ticker Group -** The NIFTY 50 stocks are well-known stocks and many of these can be grouped into stocks coming from companies in similar industries. These industries are -

- Cement
- Energy
- Information Technology
- Banking and Financial Services
- Automotive
- Metals
- Pharmaceuticals
- FMCG
- Conglomerates
- Others

Upon adding this information to the dataset, and running a Chi-Square Test of Independence between this new feature and the Target, we found out that there is absolutely no effect of this additional information in predicting the Target (as shown below).



```
Chi-Square Test Statistic: 12.112616068556745
P-Value: 0.20703476580319577
There is no significant relationship between the Ticker Group and the Target.
```

Therefore, the Stock Ticker Group feature was removed.

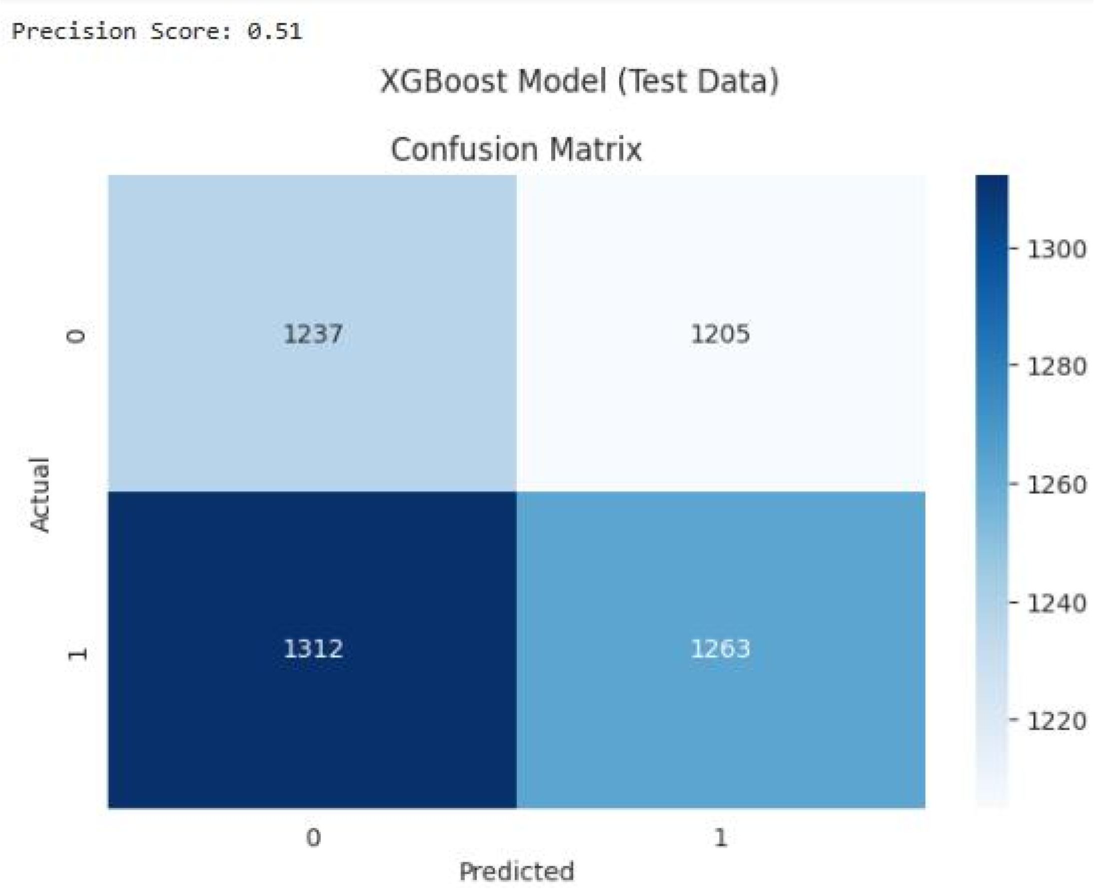
## Hyper-Parameter Tuning

Below are the hyper-parameters and their values used for the tuning process using Grid Search -

```
  'n_estimators': [50, 100, 200, 300, 400, 500, 600, 700],
 'learning_rate': [0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5],
     'max_depth': [1, 3, 5, 7, 9],
         'gamma': [0, 0.1, 0.2, 0.3, 0.4, 0.5],
     'reg_alpha': [0, 0.01, 0.1, 1, 10],
    'reg_lambda': [0, 0.01, 0.1, 1, 10]
```

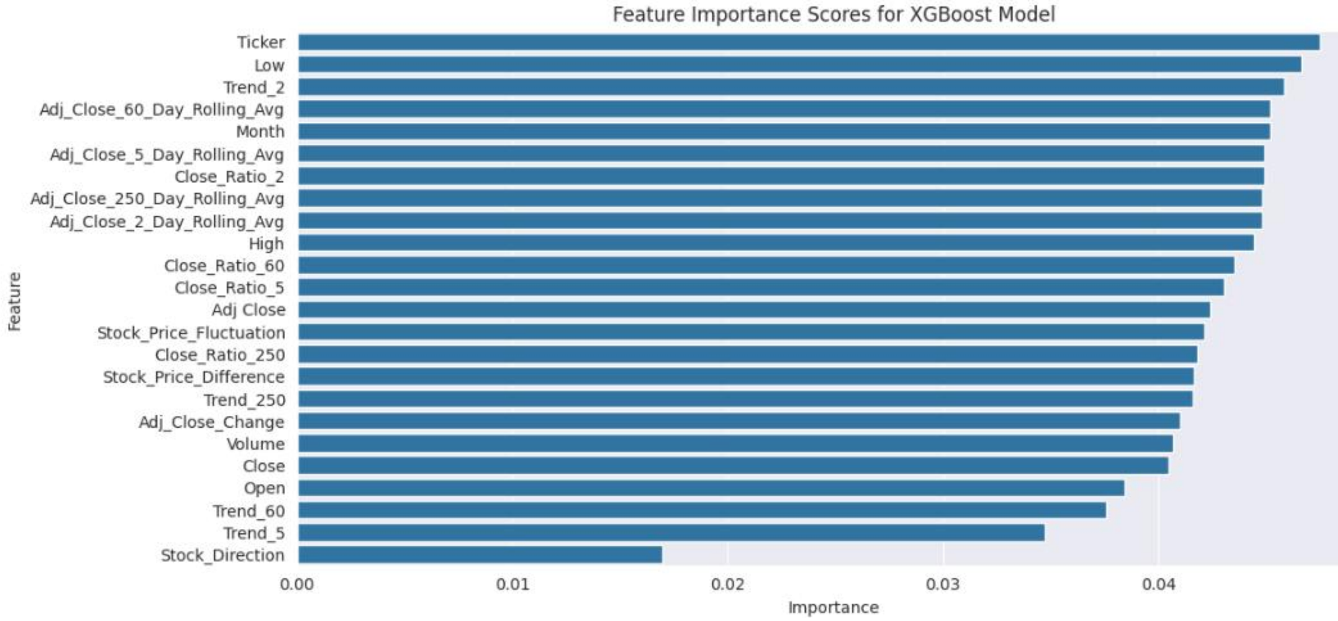
## Best Model Performance -

The best model out of the lot was an XGBoost model with **n_estimators** set to **100**, and the rest being default parameters. It achieved a **Test Precision** of **0.51**



```
Precision Score: 0.51
```

### Model Feature Importance Plot -



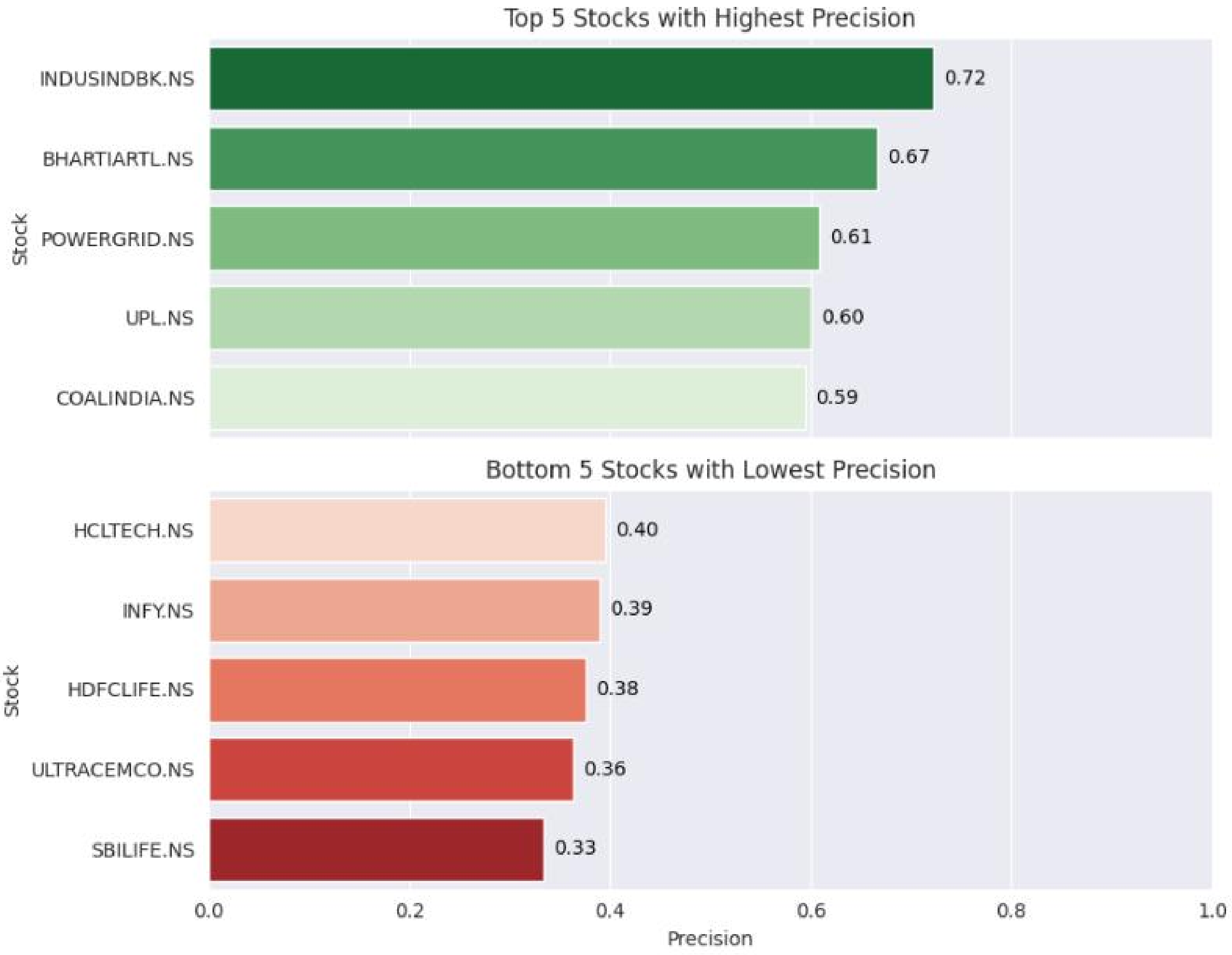
**Observations -**

- The top 5 features that contribute towards predicting the Stock Direction are:
    1. Ticker (Stock ticker)
    2. Low (Lowest price for the stock, for the current day)
    3. Trend 2 (Engineered Feature)
    4. Adj_Close_5_Day_Rolling_Avg (Engineered Feature)
    5. Month

- The Feature Importance plot suggests that many of the engineered features contribute towards the model's predictions

## Error Analysis of the Model -

To perform error analysis, we've calculated the model's precision on the test set for different Stock Tickers and then picked the top 5 and the bottom 5 in terms of test precision.



**Observations -**

- Based on the above analysis, we see that the model does well w.r.t Precision on stocks such as -
    1. **INDUSINDBK.NS (0.72)**
    2. **BHARTIARTL.NS (0.67)**
    3. **POWERGRID.NS (0.61)**

- On the other end, the model performs very poorly on stocks such as -
  1. **SBILIFE.NS (0.33)**
  2. **ULTRACEMCO.NS (0.36)**
  3. **HDFCLIFE.NS (0.38)**

## Final Take-away, Insights and Recommendations:

Predicting the stock-market is a very difficult task, even when using the most sophisticated algorithms. We've seen that improving on the base line **Precision** of **0.52** is complicated even after a good amount of Feature Engineering.

However, some important take-away from the analysis are -

- Best Stocks to invest in based on last month's performance -
  1. **M&M.NS**
  2. **HEROMOTOCO.NS**
  3. **HINDALCO.NS**

- Similarly, best stocks to invest in, based on last year's performance -
  1. **NTPC.NS**
  2. **COALINDIA.NS**
  3. **IOC.NS**

- **IOC.NS** has one of the most-cost efficient stock prices, and have grown tremendously over the last year and can be a really smart investment option

- The Test-Precision score of the model is **0.51**, which means that of all the times the model predicts that a stock is going to go up in price, **51%** of those times the model would get it right. This is lower than the baseline heuristic model by **1%**

- However, we've seen that the model's precision on the test set varies for different stock options. An example of this is having a precision score of **0.72** for **INDUSINDBK.NS** whereas having a precision score of **0.33** for **SBILIFE.NS**

- The XGBoost Model found the following features really useful for predicting the stock price movement -
  1. **Ticker (Stock ticker)**
  2. **Low (Lowest price for the stock, for the current day)**
  3. **Trend 2 (Engineered Feature)**
  4. **Adj_Close_5_Day_Rolling_Avg (Engineered Feature)**
  5. **Month**

## Next Steps to improve Predictive Performance

Predicting the movement of 49 stocks is a very complicated task. Here are some next steps that can be taken to improve the predictive performance -

- **Limiting the Scope of the Problem** - We've seen the model doing well on some stocks and terribly on others. It would make sense to have the model predict only a single stock's movement instead of 49 stocks
- **Increasing the Volume of Data** - The data is capped to only include stock data up-till 2015. We can fetch older data in order to get the model more data points so that we can reduce the over-fitting

- **Engineering Better Features** - There are more features that can help greatly to predict the movement of stock prices that are commonly used by financial analysts such as Relative Strength Index (RSI), Bollinger Bands, etc.
- **Experimenting with LSTM Models** - LSTM's are well known Recurrent Neural Networks that are capable of modeling sequence data. By collecting more data, we can leverage the power of LSTM to improve the precision score