

Stock Market

March 24, 2024

STOCK MARKET INDEX ANALYSIS

WHAT IS STOCK MARKET?

The stock market is a marketplace where investors buy and sell shares of companies. Owning a share means owning a tiny piece of that company. Trading floors or online platforms connect buyers and sellers, dictating share prices based on supply and demand. Companies raise capital by selling shares, while investors seek profit through dividends or share price appreciation. Risks are inherent: the market is volatile and prices can fluctuate.

The stock market is a marketplace where investors buy and sell shares of companies. Owning a share means owning a tiny piece of that company. **Trading floors or online platforms** connect buyers and sellers, dictating share prices based on supply and demand. **Companies raise capital** by selling shares, while **investors seek profit** through dividends or share price appreciation. Risks are inherent: the market is volatile and prices can fluctuate. Research and understanding your risk tolerance are essential. Global events and economic factors can significantly impact the market. Diversification across sectors and assets can mitigate risk. Investing in the stock market can be rewarding but requires patience and knowledge. Consider seeking professional advice before making any investments.

ABOUT NATIONAL STOCK EXCHANGE (NSE)

1. Established in 1992, the NSE is headquartered in Mumbai, India. 2. It operates as a fully automated electronic trading platform. 3. NSE facilitates trading in equities, derivatives, currencies, and debt securities, and is one of the largest stock exchanges globally.

WHAT IS NIFTY?

The Nifty represents a benchmark stock market index that is carefully selected to represent the weighted average performance of 50 carefully chosen Indian corporate stocks that are listed on the **National Stock Exchange of India (NSE)** and span a variety of sectors. For investors and traders, it is a fundamental gauge of the health and effectiveness of the Indian equity market, which is frequently praised as a true indicator of the country's economic vitality.

```
[4]: import pandas as pd
import numpy as np
nif=pd.read_csv(r"C:\Users\lenovo\Downloads\NIFTY 50-9-02-2023-to-9-02-2024.
↪csv")
nif = pd.DataFrame(nif)
print(nif.shape[0] , "Datapoints are being considered for the study of this_
↪particular analysis")
```

```
print("\nA GLANCE AT THE DATA \n")
nif['Date ']=pd.to_datetime(nif['Date '])
nif.head()
```

248 Datapoints are being considered for the study of this particular analysis

A GLANCE AT THE DATA

```
[4]:
```

	Date	Open	High	Low	Close	Shares Traded	\
0	2023-02-09	17885.50	17916.90	17779.80	17893.45	260854055	
1	2023-02-10	17847.55	17876.95	17801.00	17856.50	231991834	
2	2023-02-13	17859.10	17880.70	17719.75	17770.90	231276483	
3	2023-02-14	17840.35	17954.55	17800.05	17929.85	244512944	
4	2023-02-15	17896.60	18034.10	17853.80	18015.85	229273800	

	Turnover (Cr)
0	21529.97
1	17063.99
2	17406.31
3	20579.80
4	21457.34

```
[5]: nif.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  248 non-null   datetime64[ns]
1   Open                  248 non-null   float64
2   High                  248 non-null   float64
3   Low                   248 non-null   float64
4   Close                 248 non-null   float64
5   Shares Traded         248 non-null   int64
6   Turnover ( Cr)        248 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 13.7 KB
```

About the Dataset

The above shown dataset has been generated through the official website of NSE (<https://www.nseindia.com/reports-indices-historical-index-data>) where in all the subsequent information required for studying the NIFTY index. The table keeps a record of **Open, Close, High, Low, Shares Traded & Turnover** for each working day from 09/02/2023 to 09/02/2024.

Understanding The Terminologies.

- **Open:** Starting price of the NIFTY index for a trading session.

- **Close:** Ending price of the NIFTY index for a trading session.
- **High:** Highest price reached by the NIFTY index during a trading session.
- **Low:** Lowest price reached by the NIFTY index during a trading session.
- **Shared trades:** Trades executed by multiple investors/traders or through shared platforms.
- **Turnover:** Total value of stocks traded during a specific period, indicating market activity.

WHAT IS NIFTY IT?

NIFTY IT is a stock market index specifically tracking the performance of the **Information Technology (IT)** sector in India. Launched in 1999, it comprises 50 of the largest and most liquid IT companies listed on the National Stock Exchange (NSE).

Companies like **Infosys, TCS, Wipro, and Tech Mahindra** are prominent members, representing various IT sub-sectors like software, IT services, and consulting.

```
[6]: niftyit=pd.read_csv(r"C:\Users\lenovo\Downloads\NIFTY_
      ↪IT_Historical_PR_01012022to09022024.csv")
      print("Number of datapoints used for this analysis are", niftyit.shape[0])
      print("\nA GLANCE AT THE DATA \n")
      niftyit['Date']=pd.to_datetime(niftyit['Date'])
      niftyit.head()
```

Number of datapoints used for this analysis are 523

A GLANCE AT THE DATA

```
[6]:
```

	Index Name	Date	Open	High	Low	Close
0	NIFTY IT	2024-02-09	37891.50	38082.05	37443.75	37699.25
1	NIFTY IT	2024-02-08	37893.45	38101.50	37625.75	37851.60
2	NIFTY IT	2024-02-07	38345.40	38373.45	37734.55	37767.75
3	NIFTY IT	2024-02-06	37267.15	38286.00	37161.60	38246.30
4	NIFTY IT	2024-02-05	37441.00	37574.45	37097.95	37162.15

```
[7]: niftyit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 523 entries, 0 to 522
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Index Name  523 non-null   object
1   Date        523 non-null   datetime64[ns]
2   Open        523 non-null   float64
3   High        523 non-null   float64
4   Low         523 non-null   float64
5   Close       523 non-null   float64
dtypes: datetime64[ns](1), float64(4), object(1)
memory usage: 24.6+ KB
```

About the Dataset

The above shown dataset has been generated through the official website of NSE (<https://www.nseindia.com/reports-indices-historical-index-data>) where in all the subsequent information required for studying the NIFTY IT index. The table keeps a record of **Open, Close, High & Low** for each working day from 07/01/2022 to 09/02/2024.

1.) DATA PREPROCESSING

```
[8]: nif.isnull().sum()
```

```
[8]: Date          0
      Open          0
      High          0
      Low           0
      Close         0
      Shares Traded  0
      Turnover ( Cr) 0
      dtype: int64
```

```
[9]: niftyit.isnull().sum()
```

```
[9]: Index Name    0
      Date         0
      Open         0
      High         0
      Low          0
      Close        0
      dtype: int64
```

```
[10]: niftyit=niftyit.drop("Index Name", axis=1)
      niftyit.tail()
```

```
[10]:      Date      Open      High      Low      Close
518 2022-01-07  38140.15  38394.80  38003.15  38139.85
519 2022-01-06  38181.35  38206.60  37757.20  38009.00
520 2022-01-05  39261.55  39293.15  38409.25  38609.15
521 2022-01-04  39243.50  39446.70  38828.60  39370.70
522 2022-01-03  38752.60  39286.35  38752.55  39123.80
```

The “Index Name” column has been removed from the NIFTY IT dataset as our analysis will not be effected by that column.

2.) DATA VISUALISATION

```
[11]: import matplotlib.pyplot as plt
      import seaborn as sns
      from matplotlib.ticker import MultipleLocator
```

2.1) NIFTY 50

```
[12]: pip install altair
```

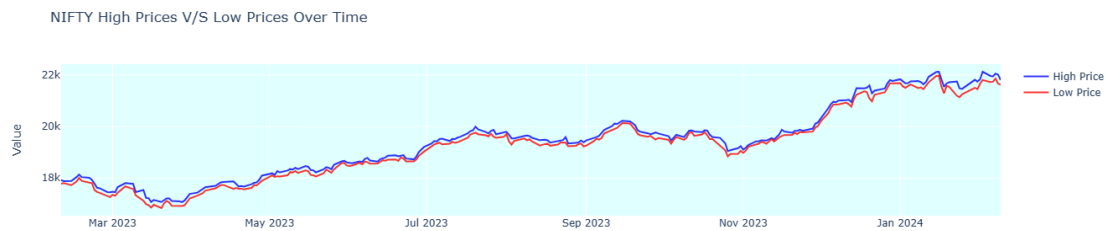
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: altair in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (5.2.0)
Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\site-packages (from altair) (22.0)
Requirement already satisfied: pandas>=0.25 in
c:\programdata\anaconda3\lib\site-packages (from altair) (1.5.3)
Requirement already satisfied: typing-extensions>=4.0.1 in
c:\programdata\anaconda3\lib\site-packages (from altair) (4.4.0)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from altair) (3.1.2)
Requirement already satisfied: jsonschema>=3.0 in
c:\programdata\anaconda3\lib\site-packages (from altair) (4.17.3)
Requirement already satisfied: toolz in c:\programdata\anaconda3\lib\site-packages (from altair) (0.12.0)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from altair) (1.23.5)
Requirement already satisfied: pyparsing!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
c:\programdata\anaconda3\lib\site-packages (from jsonschema>=3.0->altair) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in
c:\programdata\anaconda3\lib\site-packages (from jsonschema>=3.0->altair) (22.1.0)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas>=0.25->altair) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas>=0.25->altair) (2022.7)
Requirement already satisfied: MarkupSafe>=2.0 in
c:\programdata\anaconda3\lib\site-packages (from jinja2->altair) (2.1.1)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas>=0.25->altair) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```
[13]: #specifying the size of figure
import plotly.graph_objs as go
import matplotlib.pyplot as plt
import seaborn as sns
import altair as alt
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=nif['Date'],
    y=nif['High'],
    name='High Price',
    line=dict(color='blue'),
    opacity=0.8
))
fig.add_trace(go.Scatter(
```

```

x=nif['Date '],
y=nif['Low '],
name='Low Price',
line=dict(color='red'),
opacity=0.8
))
fig.update_layout(
    title="NIFTY High Prices V/S Low Prices Over Time",
    yaxis_title="Value",
    showlegend=True,
    plot_bgcolor='lightcyan'
)
fig.show()

```



```

[14]: base = alt.Chart(nif).transform_calculate(
    legend1="'Close prices of stocks'",
    legend2="'Open price of stock'",
)
scale = alt.Scale(domain=["Close prices of stocks", "Open price of stock"],
    range=['green', 'orange', ])

line1 = base.mark_line(color='green').encode(
    x = 'Date :T',
    y = 'Close :Q',
    color=alt.Color('legend1:N', scale=scale, title=''),
)

line2 = base.mark_line(color='orange').encode(
    x = 'Date :T',
    y = 'Open :Q',
    color=alt.Color('legend2:N', scale=scale, title='')
)
text = alt.Chart(nif).mark_text(
    align='left',
    baseline='middle',

```

```

        dx=3
    )
    (line1 + line2 + text).properties(
        title={"text":["Timeseries Plot of NIFTY Close and Open Price of Stock Over_
↪Year"]},
            "fontSize":20,
            "fontWeight": 'bold',
        },
        height=500, width=600
    ).interactive()

```

[14]: alt.LayerChart(...)

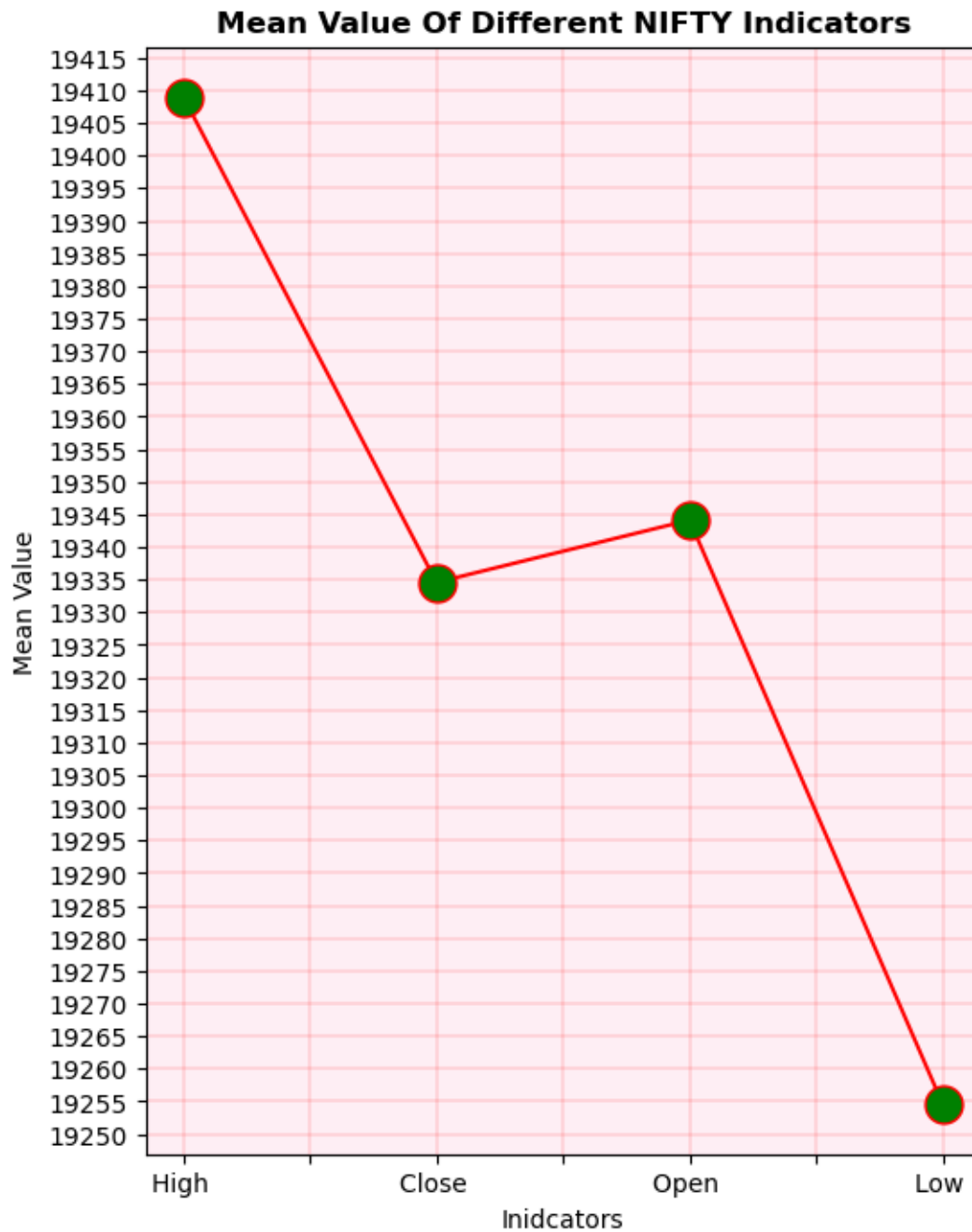
OBSERVATION

NIFTY 50 has been a very stable market when compared for the opening and closing sensex also for the high and low of the sensex points. NIFTY 50 maintained a very stable growth with very rare incidents when the differences between the opening and closing also the high and low sensex values has a huge differences. The graphs plotted shows the study where it can be clearly observed that the opening and closing sensex points for the particular duration has been stable and mainly observed a difference of 100 sensex points in average. With the clear depiction and study from graphs plotted it can be concluded that NIFTY 50 has performed in a gentle and stable manner than usual in the duration of 2 years or 24 months. The trade of stock in NIFTY 50 also didn't offer much loss or went through breakevens which could result in huge gross profits. The investors neither lost a large chunk nor gained a large chunk of money in the market, thus the performance observed was very stable for all the sensex values.

```

[15]: plt.rcParams['figure.figsize']=(6,8)
ax=plt.axes()
ax.set_facecolor("#feeef4")
mean=nif[['High ', 'Close ', 'Open ', 'Low ']].mean()
mean.plot(color="red", marker="o", markersize=15, markerfacecolor="green")
plt.grid(True, alpha= 0.5, color="red", linewidth=0.3)
plt.gca().yaxis.set_major_locator(MultipleLocator(5))
plt.xlabel("Indicators")
plt.ylabel("Mean Value")
plt.title("Mean Value Of Different NIFTY Indicators", fontweight="semibold")
plt.show()
mean.to_frame()

```



[15]:

	0
High	19408.954234
Close	19334.569153
Open	19344.116532
Low	19254.652621


```

[16]: fig = go.Figure()
fig.add_trace(go.Scatter(
    x=nif['Date '],
    y=nif['Low '],
    name='Low Price',
    line=dict(color='#dc13ff'),
    opacity=0.8
))
fig.add_trace(go.Scatter(
    x=nif['Date '],
    y=nif['High '],
    name='High Price',
    line=dict(color='blue'),
    opacity=0.8
))
fig.update_layout(
    title="DATES WITH STEEP FALL IN MARKET VALUE",
    yaxis_title="Value",
    showlegend=True,
    plot_bgcolor='#fef5b6'
)

fig.add_shape(
    type="line",
    x0='2023-03-31',
    y0=16000,
    x1='2023-03-31',
    y1=22500,
    line=dict(
        color="green",
        width=2,
        dash="solid",
    )
)

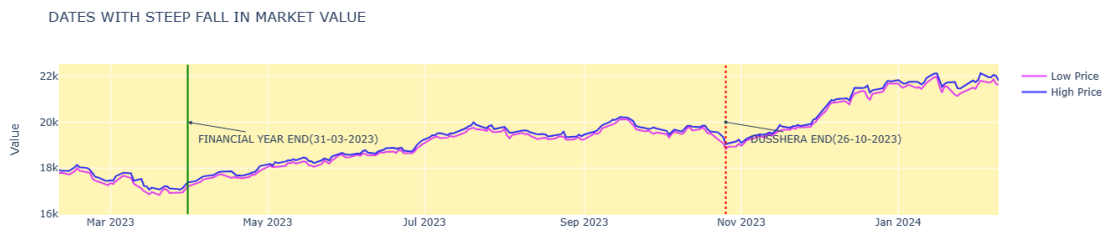
fig.add_shape(
    type="line",
    x0='2023-10-26',
    y0=16000,
    x1='2023-10-26',
    y1=22500,
    line=dict(
        color="RED",
        width=2,
        dash="dot",
    )
)

```

```

fig.add_annotation(
    x='2023-03-31',
    y=20000,
    text="FINANCIAL YEAR END(31-03-2023)",
    showarrow=True,
    arrowhead=2,
    ax=120,
    ay=20
)
fig.add_annotation(
    x='2023-10-26',
    y=20000,
    text="DUSSHERA END(26-10-2023)",
    showarrow=True,
    arrowhead=6,
    ax=120,
    ay=20
)
fig.show()

```



EXCEPTIONS

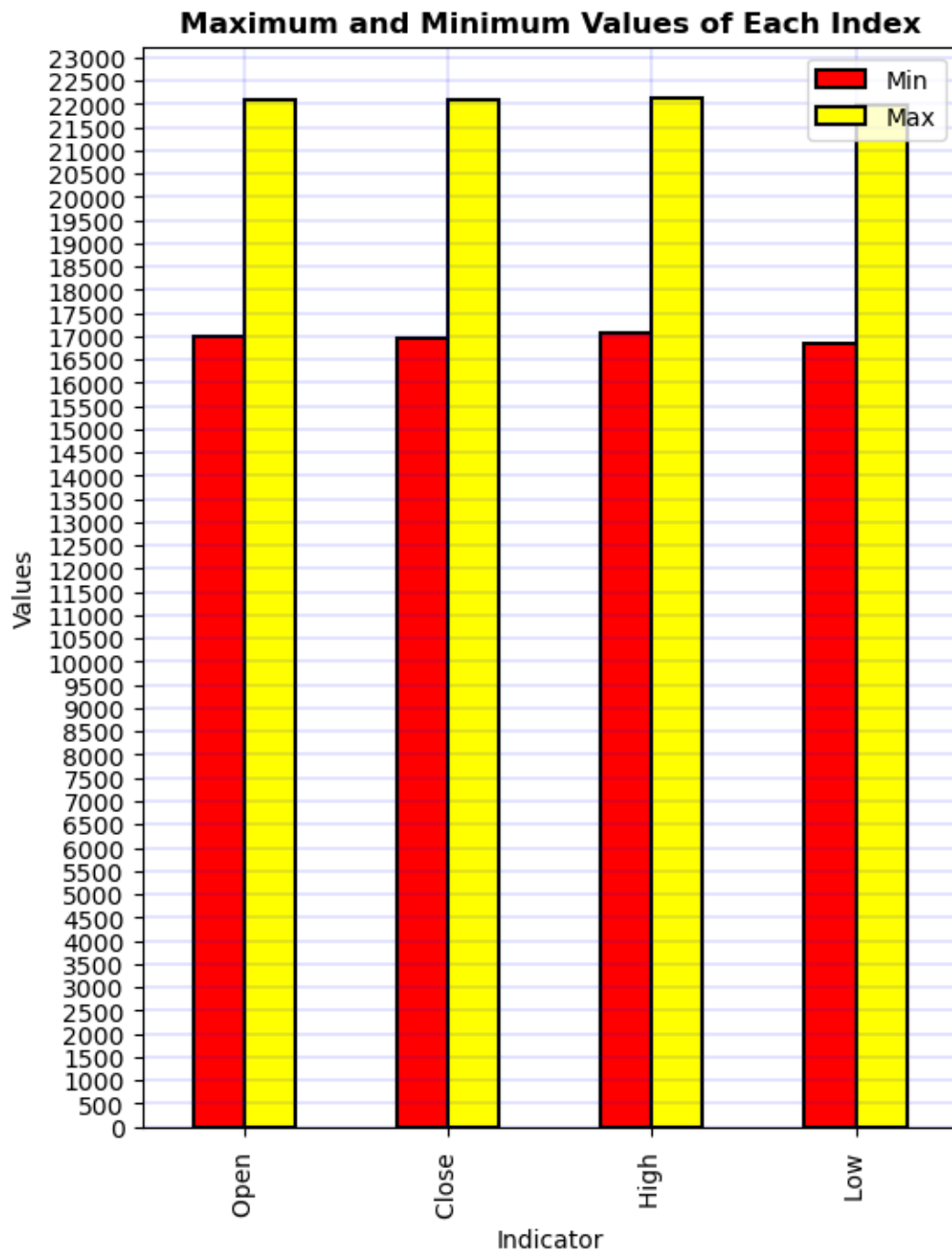
On the fiscal year end of 2023 the market saw its lowest point and since then it kept on growing till the end of Dusshera or the Festive seasons where after it experienced a drop in value. Since then the market has been constantly growing with no or little dip in valuation.

```

[17]: max_values = nif[['Open ', 'Close ', 'High ', 'Low ']].max()
min_values = nif[['Open ', 'Close ', 'High ', 'Low ']].min()
bar_df = pd.DataFrame({'Min': min_values, 'Max': max_values})
bar_df.plot(kind='bar', color=['red', 'yellow'], edgecolor=['black', 'black'],
    linewidth=1.5)
plt.grid(True, alpha= 0.5, color="blue", linewidth=0.3)
plt.gca().yaxis.set_major_locator(MultipleLocator(500))
plt.title('Maximum and Minimum Values of Each Index', fontweight="semibold")
plt.xlabel('Indicator')
plt.ylabel('Values')

```

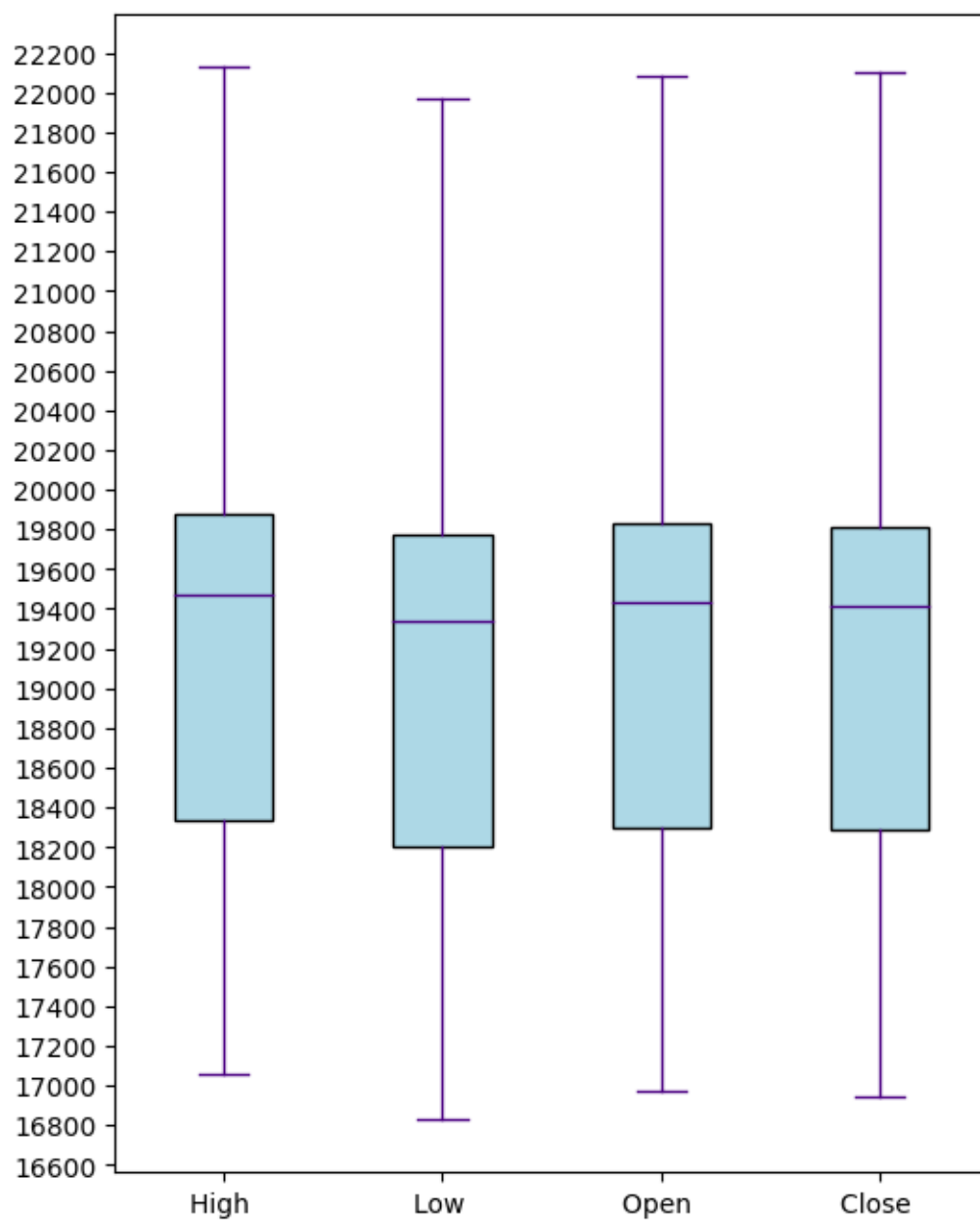
```
plt.show()  
bar_df=pd.DataFrame(bar_df)  
bar_df
```



```
[17]:
```

	Min	Max
Open	16977.30	22080.50
Close	16945.05	22097.45
High	17061.75	22126.80
Low	16828.35	21969.80

```
[18]: nif[['High ', 'Low ', 'Open ', 'Close ']].plot(kind='box',
        color='Indigo', patch_artist=True, boxprops=dict(facecolor='lightblue'))
plt.gca().yaxis.set_major_locator(MultipleLocator(200))
plt.show()
```



BOX PLOT REPRESENTATION

A box plot, also known as a box-and-whisker plot, is a graphic depiction of a dataset's distribution that sheds light on its outlier prevalence, central tendency, and dispersion. The interquartile range (IQR), or the place where the centre 50% of the data lies, is represented by a rectangular box on the plot, and the median value is indicated by a line. Outliers, if any, are displayed separately, and whiskers extend from the box to the minimum and maximum values within 1.5 times the IQR. Box plots provide a succinct overview of important statistical metrics and are helpful for comparing distributions and locating anomalies in datasets.

2.2) NIFTY IT

```
[19]: fig=go.Figure()
fig.add_trace(go.Candlestick(
    x=niftyit['Date'],
    open=niftyit['Open'],
    low=niftyit['Low'],
    high=niftyit['High'],
    close=niftyit['Close'],
    name='Low Price',
    increasing=dict(line=dict(color='green')),
    decreasing=dict(line=dict(color='red'))
))
fig.update_layout(
    title="Candlestick Chart",
    yaxis_title="Value",
    xaxis_title="Date",
    showlegend=True,
    plot_bgcolor='#e2baff',
    xaxis_rangeslider_visible=True
)
```



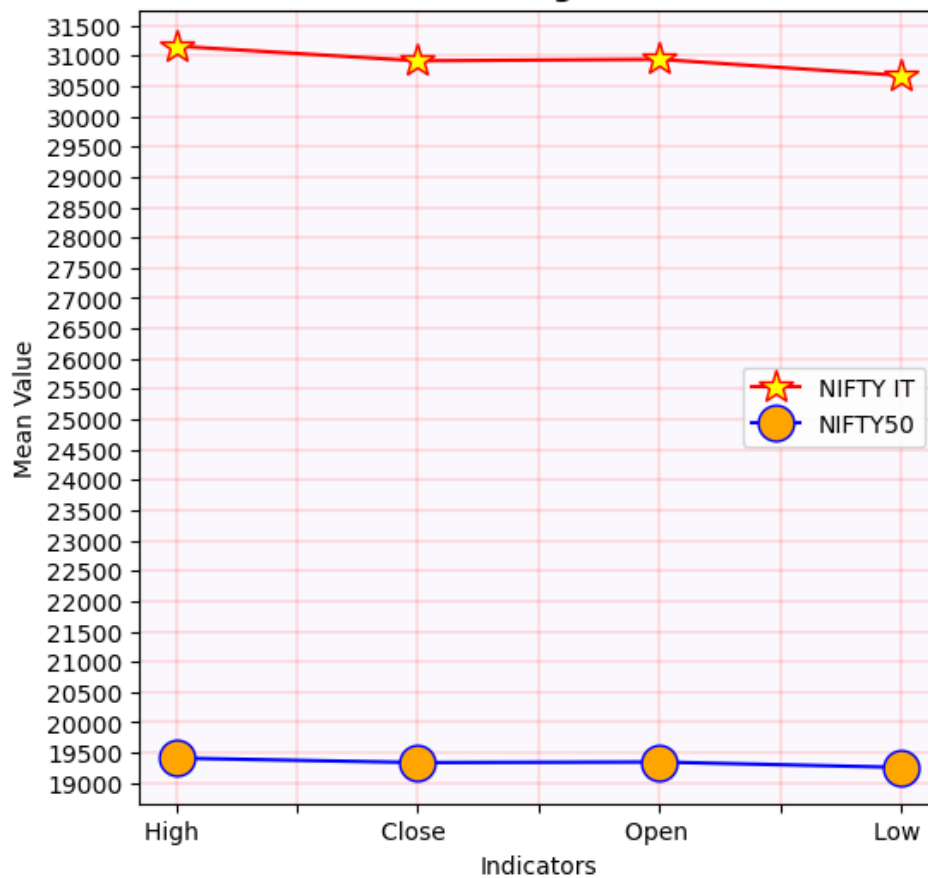
```
[20]: mean1=niftyit[['High', 'Close', 'Open', 'Low']].mean()
plt.rcParams['figure.figsize']=(6,6)
ax=plt.axes()
```

```

ax.set_facecolor("#fbf8fd")
mean1.plot(color="RED",marker="*", markersize=15, markerfacecolor="yellow",
↪label="NIFTY IT")
mean.plot(color="blue",marker="o", markersize=15, markerfacecolor="orange",
↪label="NIFTY50")
plt.grid(True, alpha= 0.5, color="red", linewidth=0.3)
plt.gca().yaxis.set_major_locator(MultipleLocator(500))
plt.legend(loc="center right")
plt.xlabel("Indicators")
plt.ylabel("Mean Value")
plt.title("The Difference Between The Average Values of NIFTY IT & NIFTY 50",
↪fontweight='semibold')
plt.show()

```

The Difference Between The Average Values of NIFTY IT & NIFTY 50



```

[21]: print("\nMean of NIFTY IT Index")
mean1.to_frame()

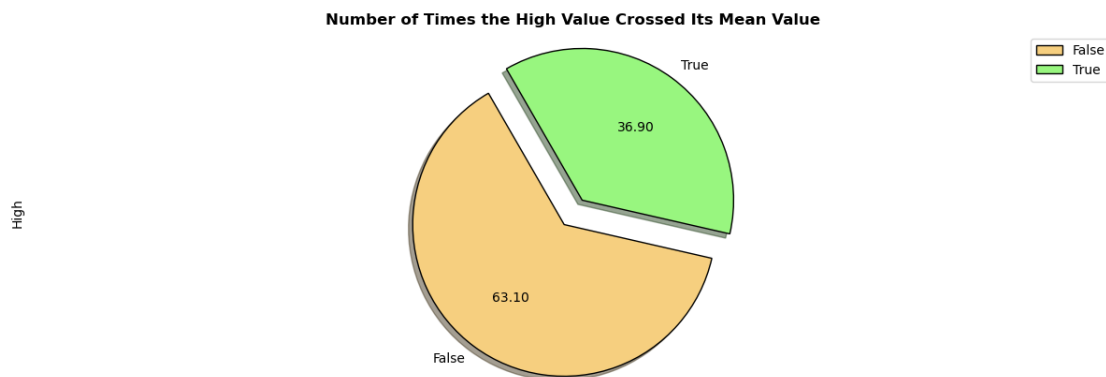
```

Mean of NIFTY IT Index

```
[21]:
      0
High   31159.160421
Close  30916.247419
Open   30940.293212
Low    30674.196558
```

We can infer from the analysis and data that NIFTY IT index averages are much higher than the NIFTY 50 index averages with mean difference of 11500.

```
[22]: plt.rcParams['figure.figsize']=(15,5)
means=(niftyit['High']>niftyit['High'].mean()).value_counts()
means.plot(kind="pie",autopct='%2f', wedgeprops={'edgecolor':"black"},
           ↪startangle=120,explode=[0.1,0.1], shadow=True , colors=["#f6cf7f","#98f67f"])
plt.axis("equal")
plt.legend(loc='upper right')
plt.title("Number of Times the High Value Crossed Its Mean Value",
           ↪fontweight="bold")
plt.show()
```

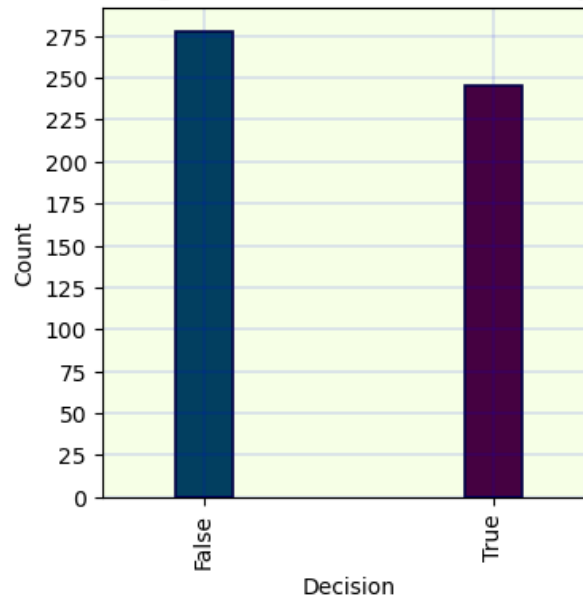


It is observed from the above pie chart that about 63% times the NIFTY IT High index stays below its average High index.

```
[23]: plt.rcParams['figure.figsize']=(4,4)
ax=plt.axes()
ax.set_facecolor("#f6ffe6")
(niftyit['Close']>niftyit['Open']).value_counts().plot(kind='bar',
           ↪color=['#023f60','#450141'], linewidth=1.2, edgecolor='#010145',width=0.2)
plt.ylabel("Count")
plt.xlabel("Decision")
plt.title("Number of Times Closing Value Was Greater Than Open Value In NIFTY_
           ↪IT", fontweight="semibold")
plt.gca().yaxis.set_major_locator(MultipleLocator(25))
```

```
plt.grid(True, alpha= 0.5, color="blue", linewidth=0.3)
plt.show()
```

Number of Times Closing Value Was Greater Than Open Value In NIFTY IT



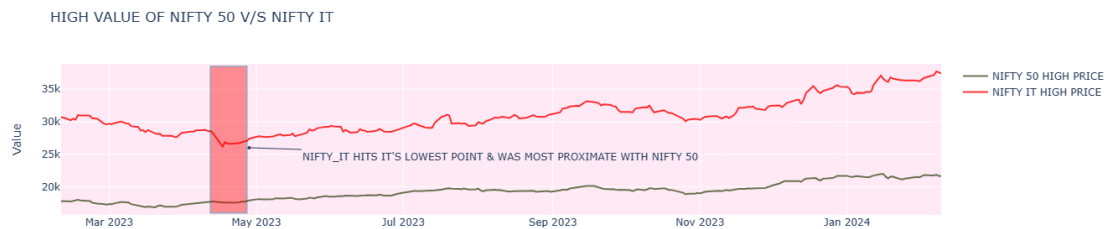
```
[24]: niftyit1=niftyit[niftyit['Date']>'2023-02-08']
niftyit1
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=nif['Date '],
    y=nif['Low '],
    name='NIFTY 50 HIGH PRICE',
    line=dict(color='#455429'),
    opacity=0.8
))
fig.add_trace(go.Scatter(
    x=niftyit1['Date'],
    y=niftyit1['Low'],
    name='NIFTY IT HIGH PRICE',
    line=dict(color='RED'),
    opacity=0.8
))
fig.update_layout(
    title="HIGH VALUE OF NIFTY 50 V/S NIFTY IT",
    yaxis_title="Value",
    showlegend=True,
    plot_bgcolor='#fee9f5'
```



```

)
fig.add_shape(
    type='rect',
    x0='2023-04-12',
    y0=16000,
    x1='2023-04-27',
    y1=38500,
    fillcolor="red",
    opacity=0.4,
)
fig.add_annotation(
    x='2023-04-28',
    y=26000,
    text="NIFTY_IT HITS IT'S LOWEST POINT & WAS MOST PROXIMATE WITH NIFTY 50",
    showarrow=True,
    arrowhead=6,
    ax=300,
    ay=10
)
)

```



REASON FOR THESE FALLS?

- **Currency Fluctuations:** Since many IT companies derive a significant portion of their revenue from overseas markets, fluctuations in currency exchange rates can affect their earnings. A strengthening domestic currency relative to major foreign currencies can impact the competitiveness of Indian IT firms and lead to a decline in their stock prices and during this phase INR was very weak in comparison to USD.
- **Global Economic Conditions:** Economic downturns or uncertainties, such as slowdowns in major economies or geopolitical tensions, can impact the IT sector's performance. Reduced business spending on technology services and products due to economic uncertainties can lead to a decline in IT stocks.
- **Demand & Supply:** One demand-supply factor that can contribute to the fall of the NIFTY IT index is a decrease in demand for IT products and services. This decline in demand may stem from economic downturns, where businesses and consumers reduce spending on technology due to budget constraints or uncertainty about future growth prospects. When demand weakens, IT companies may experience lower sales volumes, reduced contract renewals, and

downward pressure on pricing, ultimately leading to a decrease in revenue and profitability. As a result, investor sentiment towards the IT sector may sour, causing the NIFTY IT index to decline. It was also due to the shortage of Integrated Chip Imports.

EFFECTS FOR THESE FALLS?

A series of economic events could culminate in a recession that is linked to the decline of the NIFTY IT index. Less business investment in technology, job losses, and lower consumer expenditure on IT-related goods and services might result from a large dip in the NIFTY IT index, which indicates worse demand and profitability in the IT sector. This decline in IT spending has an impact on suppliers, service providers, and other technology-dependent auxiliary firms across a range of industries. Employment, financial markets, and consumer confidence are all affected, which exacerbates economic downturns and may even set off a wider recessionary climate marked by decreased GDP, growing unemployment, and decreased economic activity

A Deep Dive Into Few Best Performing NIFTY IT Stocks

```
[25]: infy=pd.read_csv(r"C:\Users\lenovo\Downloads\Quote-Equity-INFY-EQ-01-01-2023-to-01-01-2024.csv")
      wip=pd.read_csv(r"C:\Users\lenovo\Downloads\Quote-Equity-WIPRO-EQ-01-01-2023-to-01-01-2024.csv")
      tcs=pd.read_csv(r"C:\Users\lenovo\Downloads\Quote-Equity-TCS-EQ-01-01-2023-to-01-01-2024.csv")
      infy['Date ']=pd.to_datetime(infy['Date '])
      wip['Date ']=pd.to_datetime(wip['Date '])
      tcs['Date ']=pd.to_datetime(tcs['Date '])
```

Analysing the VWAP Metric

VWAP stands for Volume Weighted Average Price. It is a trading metric used in finance, particularly in the context of stocks and securities trading. VWAP represents the average price at which a stock has traded throughout the day, based on both volume and price.

Here's how VWAP is calculated:

- For each trade that occurs during the trading day, multiply the price of the trade by the volume (number of shares) traded.
- Sum up all of these price-volume products.
- Divide this sum by the total volume of shares traded throughout the day.

The result is the VWAP, which gives a sense of the average price at which the stock has traded over the entire trading day, weighted by the volume of each trade. **Traders often use VWAP as a benchmark to assess the effectiveness of their trading strategies.** They may compare the current price of a stock to its VWAP to gauge whether the stock is currently trading above or below its average price for the day. Additionally, VWAP is used by institutional traders for executing large orders in a way that minimizes market impact.

```
[26]: fig = go.Figure()
      fig.add_trace(go.Scatter(
          x=wip['Date '],
```

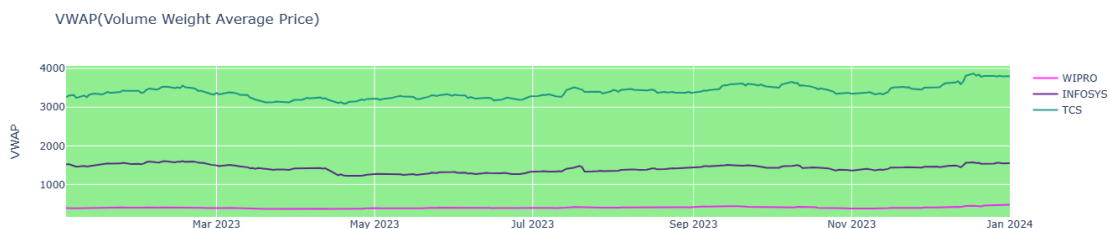
```

        y=wip['vwap '],
        name='WIPRO',
        line=dict(color='magenta'),
        opacity=0.8
    ))
    fig.add_trace(go.Scatter(
        x=infy['Date '],
        y=infy['vwap '],
        name='INFOSYS',
        line=dict(color='indigo'),
        opacity=0.8
    ))

    fig.add_trace(go.Scatter(
        x=tcs['Date '],
        y=tcs['vwap '],
        name='TCS',
        line=dict(color='teal'),
        opacity=0.8
    ))

    fig.update_layout(
        title="VWAP(Volume Weight Average Price)",
        yaxis_title="VWAP",
        showlegend=True,
        plot_bgcolor='lightgreen'
    )
    fig.show()

```



FUTURE SCOPE

Time Series Forecasting of any stock using models like:

Facebook Prophet(fbprohet)

Arima(Auto Regressive Integrated Moving Average)

LSTM(Long Short Term Memory)

[27]: !pip install keras

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: keras in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (3.1.1)
Requirement already satisfied: rich in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(13.7.1)
Requirement already satisfied: absl-py in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(2.1.0)
Requirement already satisfied: namex in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(0.0.7)
Requirement already satisfied: ml-dtypes in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(0.3.2)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-
packages (from keras) (1.23.5)
Requirement already satisfied: optree in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(0.10.0)
Requirement already satisfied: h5py in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras)
(3.10.0)
Requirement already satisfied: typing-extensions>=4.0.0 in
c:\programdata\anaconda3\lib\site-packages (from optree->keras) (4.4.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from
rich->keras) (2.17.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from
rich->keras) (3.0.0)
Requirement already satisfied: mdurl~=0.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from markdown-
it-py>=2.2.0->rich->keras) (0.1.2)
```

[28]: !pip install tensorflow

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: tensorflow in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (2.16.1)
Requirement already satisfied: tensorflow-intel==2.16.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow)
(2.16.1)
Requirement already satisfied: wrapt>=1.11.0 in
c:\programdata\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.14.1)
```

Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.5.4)

Requirement already satisfied: astunparse>=1.6.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.6.3)

Requirement already satisfied:
protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.0dev,>=3.20.3
in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from
tensorflow-intel==2.16.1->tensorflow) (4.25.3)

Requirement already satisfied: ml-dtypes~=0.3.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.3.2)

Requirement already satisfied: typing-extensions>=3.6.6 in
c:\programdata\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (4.4.0)

Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\site-
packages (from tensorflow-intel==2.16.1->tensorflow) (22.0)

Requirement already satisfied: requests<3,>=2.21.0 in
c:\programdata\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.28.1)

Requirement already satisfied: opt-einsum>=2.3.2 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (3.3.0)

Requirement already satisfied: libclang>=13.0.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (18.1.1)

Requirement already satisfied: grpcio<2.0,>=1.24.3 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.62.1)

Requirement already satisfied: keras>=3.0.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (3.1.1)

Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
c:\programdata\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (1.23.5)

Requirement already satisfied: flatbuffers>=23.5.26 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (24.3.7)

Requirement already satisfied: absl-py>=1.0.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (2.1.0)

Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.31.0)

Requirement already satisfied: google-pasta>=0.1.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.2.0)

Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1->tensorflow) (65.6.3)

Requirement already satisfied: h5py>=3.10.0 in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-intel==2.16.1->tensorflow) (3.10.0)

Requirement already satisfied: termcolor>=1.1.0 in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-intel==2.16.1->tensorflow) (2.4.0)

Requirement already satisfied: tensorboard<2.17,>=2.16 in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorflow-intel==2.16.1->tensorflow) (2.16.2)

Requirement already satisfied: six>=1.12.0 in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1->tensorflow) (1.16.0)

Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\programdata\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-intel==2.16.1->tensorflow) (0.38.4)

Requirement already satisfied: optree in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.10.0)

Requirement already satisfied: namex in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.0.7)

Requirement already satisfied: rich in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (13.7.1)

Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (2.0.4)

Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (2024.2.2)

Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (3.4)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.16.1->tensorflow) (1.26.14)

Requirement already satisfied: werkzeug>=1.0.1 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (2.2.2)

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\lenovo\appdata\roaming\python\python310\site-packages (from tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (0.7.2)

Requirement already satisfied: markdown>=2.6.8 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (3.4.1)

Requirement already satisfied: MarkupSafe>=2.1.1 in c:\programdata\anaconda3\lib\site-packages (from

```
werkzeug>=1.0.1->tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow)
(2.1.1)
```

```
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from
rich->keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (2.17.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from
rich->keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (3.0.0)
Requirement already satisfied: mdurl~=0.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from markdown-
it-py>=2.2.0->rich->keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.1.2)
```

```
[29]: import tensorflow as tf
      from tensorflow import keras
      from tensorflow.keras import layers
```

```
[30]: data=pd.read_csv(r"C:\Users\lenovo\Downloads\TCS.NS.csv")
      data
```

```
[30]:
```

	Date	Open	High	Low	Close \
0	2022-02-08	3791.550049	3799.699951	3721.000000	3743.449951
1	2022-02-09	3750.000000	3777.949951	3746.100098	3760.550049
2	2022-02-10	3789.800049	3789.800049	3758.000000	3770.350098
3	2022-02-11	3752.500000	3752.500000	3690.000000	3694.949951
4	2022-02-14	3724.000000	3793.250000	3710.000000	3733.750000
..
489	2024-02-01	3820.000000	3904.899902	3805.050049	3854.149902
490	2024-02-02	3875.050049	3984.399902	3872.000000	3966.300049
491	2024-02-05	3983.000000	4022.000000	3962.100098	3973.300049
492	2024-02-06	3997.000000	4149.899902	3990.000000	4136.000000
493	2024-02-07	4153.000000	4155.000000	4073.949951	4083.399902

	Adj Close	Volume
0	3563.721436	2946443
1	3580.000488	1638721
2	3589.329834	2112459
3	3517.549805	3851488
4	3554.487061	5951745
..
489	3854.149902	2363107
490	3966.300049	2826510
491	3973.300049	1691523
492	4136.000000	4474396
493	4083.399902	2124267

```
[494 rows x 7 columns]
```

MODEL SELECTION

```
[40]: from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1, 1))

train_size = int(len(scaled_data) * 0.8)
test_size = len(scaled_data) - train_size
train_data, test_data = scaled_data[0:train_size, :], scaled_data[train_size:
    ↪len(scaled_data), :]

model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(train_data.
    ↪shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))

model.compile(optimizer='adam', loss='mean_squared_error')

model.fit(train_data, train_data, epochs=50, batch_size=32)

test_predict = model.predict(test_data)
test_predict = scaler.inverse_transform(test_predict)
```

C:\Users\lenovo\AppData\Roaming\Python\Python310\site-packages\keras\src\layers\rnn\rnn.py:204: UserWarning:

Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```
Epoch 1/50
13/13          2s 2ms/step - loss:
0.1086
Epoch 2/50
13/13          0s 3ms/step - loss:
0.0805
Epoch 3/50
13/13          0s 3ms/step - loss:
0.0382
Epoch 4/50
13/13          0s 3ms/step - loss:
0.0167
Epoch 5/50
```


13/13	0s 3ms/step - loss:
0.0146	
Epoch 6/50	
13/13	0s 3ms/step - loss:
0.0121	
Epoch 7/50	
13/13	0s 3ms/step - loss:
0.0116	
Epoch 8/50	
13/13	0s 2ms/step - loss:
0.0100	
Epoch 9/50	
13/13	0s 3ms/step - loss:
0.0094	
Epoch 10/50	
13/13	0s 3ms/step - loss:
0.0064	
Epoch 11/50	
13/13	0s 3ms/step - loss:
0.0065	
Epoch 12/50	
13/13	0s 3ms/step - loss:
0.0041	
Epoch 13/50	
13/13	0s 3ms/step - loss:
0.0034	
Epoch 14/50	
13/13	0s 3ms/step - loss:
0.0020	
Epoch 15/50	
13/13	0s 2ms/step - loss:
0.0014	
Epoch 16/50	
13/13	0s 3ms/step - loss:
0.0014	
Epoch 17/50	
13/13	0s 2ms/step - loss:
0.0010	
Epoch 18/50	
13/13	0s 3ms/step - loss:
0.0015	
Epoch 19/50	
13/13	0s 6ms/step - loss:
0.0011	
Epoch 20/50	
13/13	0s 3ms/step - loss:
0.0010	
Epoch 21/50	

13/13	0s 3ms/step - loss:
0.0011	
Epoch 22/50	
13/13	0s 4ms/step - loss:
0.0012	
Epoch 23/50	
13/13	0s 4ms/step - loss:
0.0012	
Epoch 24/50	
13/13	0s 3ms/step - loss:
0.0010	
Epoch 25/50	
13/13	0s 3ms/step - loss:
0.0013	
Epoch 26/50	
13/13	0s 3ms/step - loss:
0.0011	
Epoch 27/50	
13/13	0s 4ms/step - loss:
0.0012	
Epoch 28/50	
13/13	0s 4ms/step - loss:
9.4523e-04	
Epoch 29/50	
13/13	0s 3ms/step - loss:
0.0011	
Epoch 30/50	
13/13	0s 3ms/step - loss:
0.0012	
Epoch 31/50	
13/13	0s 4ms/step - loss:
0.0012	
Epoch 32/50	
13/13	0s 4ms/step - loss:
0.0012	
Epoch 33/50	
13/13	0s 4ms/step - loss:
0.0010	
Epoch 34/50	
13/13	0s 3ms/step - loss:
9.9207e-04	
Epoch 35/50	
13/13	0s 4ms/step - loss:
0.0010	
Epoch 36/50	
13/13	0s 3ms/step - loss:
8.2642e-04	
Epoch 37/50	

```

13/13          0s 4ms/step - loss:
9.1993e-04
Epoch 38/50
13/13          0s 2ms/step - loss:
9.5775e-04
Epoch 39/50
13/13          0s 3ms/step - loss:
0.0013
Epoch 40/50
13/13          0s 3ms/step - loss:
8.0426e-04
Epoch 41/50
13/13          0s 5ms/step - loss:
8.2983e-04
Epoch 42/50
13/13          0s 3ms/step - loss:
0.0011
Epoch 43/50
13/13          0s 3ms/step - loss:
0.0011
Epoch 44/50
13/13          0s 3ms/step - loss:
0.0011
Epoch 45/50
13/13          0s 3ms/step - loss:
0.0010
Epoch 46/50
13/13          0s 4ms/step - loss:
7.9707e-04
Epoch 47/50
13/13          0s 3ms/step - loss:
0.0010
Epoch 48/50
13/13          0s 2ms/step - loss:
6.0661e-04
Epoch 49/50
13/13          0s 3ms/step - loss:
0.0011
Epoch 50/50
13/13          0s 3ms/step - loss:
8.4694e-04
4/4           1s 127ms/step

```

```
[47]: model.summary()
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 1, 50)	10,400
dropout_2 (Dropout)	(None, 1, 50)	0
lstm_3 (LSTM)	(None, 50)	20,200
dropout_3 (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 1)	51

Total params: 91,955 (359.20 KB)

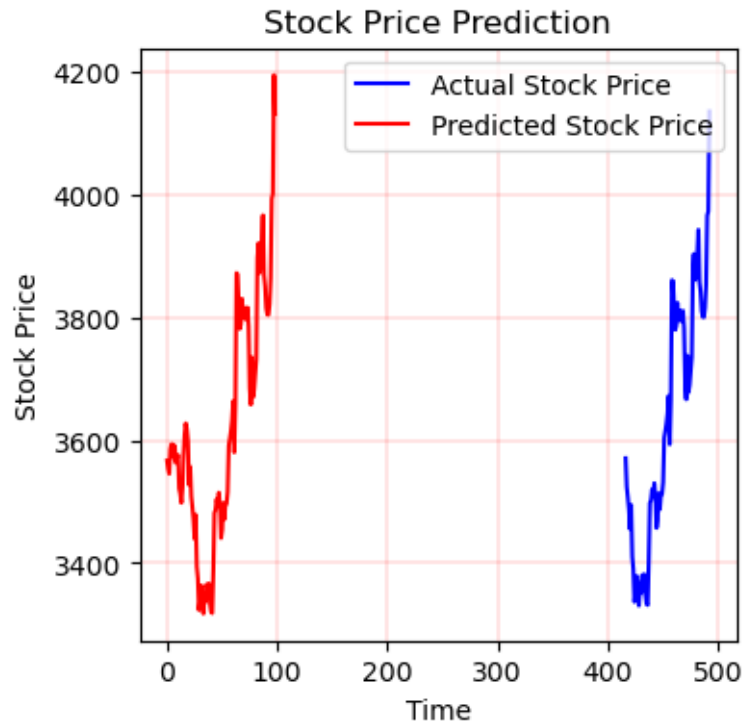
Trainable params: 30,651 (119.73 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 61,304 (239.47 KB)

MODEL EVALUATION

```
[48]: plt.grid(True, alpha= 0.5, color="red", linewidth=0.3)
plt.plot(data['Close'][train_size+21:len(data)-1 :], color='blue',
        label='Actual Stock Price')
plt.plot(test_predict, color='red', label='Predicted Stock Price')
plt.title('Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend(loc='upper right')
plt.show()
```



```
[32]: !pip install nbconvert nbformat
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nbconvert in c:\programdata\anaconda3\lib\site-
packages (6.5.4)
Requirement already satisfied: nbformat in c:\programdata\anaconda3\lib\site-
packages (5.7.0)
Requirement already satisfied: bleach in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (4.1.0)
Requirement already satisfied: jupyterlab-pygments in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: entrypoints>=0.2.2 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (0.4)
Requirement already satisfied: jupyter-core>=4.7 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (5.2.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (1.5.0)
Requirement already satisfied: mistune<2,>=0.8.1 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: tinycss2 in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (1.2.1)
Requirement already satisfied: beautifulsoup4 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (4.11.1)
Requirement already satisfied: MarkupSafe>=2.0 in
```

c:\programdata\anaconda3\lib\site-packages (from nbconvert) (2.1.1)
Requirement already satisfied: nbclient>=0.5.0 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (0.5.13)
Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (22.0)
Requirement already satisfied: jinja2>=3.0 in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (3.1.2)
Requirement already satisfied: pygments>=2.4.1 in
c:\users\lenovo\appdata\roaming\python\python310\site-packages (from nbconvert)
(2.17.2)
Requirement already satisfied: defusedxml in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (0.7.1)
Requirement already satisfied: traitlets>=5.0 in
c:\programdata\anaconda3\lib\site-packages (from nbconvert) (5.7.1)
Requirement already satisfied: lxml in c:\programdata\anaconda3\lib\site-
packages (from nbconvert) (4.9.1)
Requirement already satisfied: fastjsonschema in
c:\programdata\anaconda3\lib\site-packages (from nbformat) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in
c:\programdata\anaconda3\lib\site-packages (from nbformat) (4.17.3)
Requirement already satisfied: attrs>=17.4.0 in
c:\programdata\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat)
(22.1.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
c:\programdata\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat)
(0.18.0)
Requirement already satisfied: platformdirs>=2.5 in
c:\programdata\anaconda3\lib\site-packages (from jupyter-core>=4.7->nbconvert)
(2.5.2)
Requirement already satisfied: pywin32>=1.0 in
c:\programdata\anaconda3\lib\site-packages (from jupyter-core>=4.7->nbconvert)
(305.1)
Requirement already satisfied: nest-asyncio in
c:\programdata\anaconda3\lib\site-packages (from nbclient>=0.5.0->nbconvert)
(1.5.6)
Requirement already satisfied: jupyter-client>=6.1.5 in
c:\programdata\anaconda3\lib\site-packages (from nbclient>=0.5.0->nbconvert)
(7.3.4)
Requirement already satisfied: soupsieve>1.2 in
c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4->nbconvert)
(2.3.2.post1)
Requirement already satisfied: webencodings in
c:\programdata\anaconda3\lib\site-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: six>=1.9.0 in c:\programdata\anaconda3\lib\site-
packages (from bleach->nbconvert) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\programdata\anaconda3\lib\site-packages (from jupyter-
client>=6.1.5->nbclient>=0.5.0->nbconvert) (2.8.2)

Requirement already satisfied: pyzmq>=23.0 in c:\programdata\anaconda3\lib\site-packages (from jupyter-client>=6.1.5->nbclient>=0.5.0->nbconvert) (23.2.0)
Requirement already satisfied: tornado>=6.0 in
c:\programdata\anaconda3\lib\site-packages (from jupyter-client>=6.1.5->nbclient>=0.5.0->nbconvert) (6.1)