# *INDUSTRIAL TRAINING REPORT*

***On***

***Project:*** ***"Data Analysis and Visualization on NIFTY 50 Dataset in Python"***

# *At*

# *ACADEMY OF SKILL DEVELOPMENT*

***Submitted in partial fulfilment of the requirement***

***For the award degree of***

***Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence & Machine Learning)***


***Submitted by:***

***SOURISH CHATTERJEE***

***Roll NO.- 14230822056***




***Meghnad Saha Institute of Technology***

***Nazirabad, Utchepota, Kolkata - 700150.***


***Affiliated To***

***Maulana Abul Kalam Azad University of Technology***

# ***DECLARATION***

*I, Sourish Chatterjee, a student pursuing Bachelor of Technology in Computer Science and Engineering* (Artificial Intelligence & Machine Learning), *hereby declare that the project titled "**Data Analysis and Visualization on NIFTY 50 Dataset in Python**" is the result of our own team work. The project has been undertaken as a part of my academic curriculum, and all the information, methodologies, and findings presented in this project are authentic and original.*
*I further declare that the dataset utilized in this project, has been appropriately sourced and acknowledged. Any external resources, including code snippets, libraries, or frameworks, have been duly credited in the project documentation.*

*Date: 08.02.2024*
*Sourish Chatterjee*
*Makaut Roll No.- 14230822056*

# ***ACKNOWLEDGEMENT***

*I would like to express my special thanks of gratitude to my teachers who gave me the opportunity to do this wonderful project on the topic* ***"Data Analysis and Visualization of NIFTY 50 Dataset",*** *which also helped me in doing a lot of research and I came to know about so many new things, I am really thankful to them.*
*Furthermore, I would also like to thank my parents, friends and teammates who helped me a lot in finalizing this project within the limited time frame.*

*My other teammates include:*

- *Dhiman Majumder (Makaut Roll -14230822025)*
- *Dwip Dutta (Makaut Roll -14230822007)*
- *Kanchan Pramanik (Makaut Roll -14230822013)*
- *Ritam Ghosh (Makaut Roll -14230822023)*

*Thanking You,*
*SOURISH CHATTERJEE*
*Makaut Roll No.- 14230822056*
*B. TECH(CSE-AIML)*

# ***ABSTRACT***

*Through this project, I aim to achieve the following objectives:*

- *Explore the trends and patterns in the NIFTY 50 dataset.*
- *Perform statistical analysis to understand the distribution and volatility of the index.*
- *Implement various data visualization techniques to present insights in a clear and interpretable manner.*
- *Provide actionable insights for investors and traders based on the analysis.*

*The outcomes of the data are effectively communicated through visual representations.The project goes beyond static visualizations by incorporating interactive elements using tools like Matplotlib, Seaborn, to enhance engagement and comprehension.*

*In addition to the technical aspects, a strong emphasis is placed on documentation. The code is thoroughly annotated, elucidating key steps and rationale behind decisions. The abstracted insights and findings are presented coherently, ensuring clarity and replicability. This project not only demonstrates technical proficiency in Python and data analysis but also prioritizes effective communication of results through compelling visualizations and documentation.*

# ***CONTENTS***

# ***PROJECT REPORT***

## ***Data Analysis and Visualization on NIFTY 50 Dataset in Python***

## I. Introduction

*In today's dynamic financial landscape, data analysis and visualization play pivotal roles in understanding market trends, making informed investment decisions, and gaining insights into the behaviour of financial instruments. The* ***NIFTY 50*** *index, comprising the top 50 companies listed on the* ***National Stock Exchange of India (NSEI****), serves as a barometer for the Indian equity market. Analysing and visualizing the historical performance of the NIFTY 50 dataset can provide invaluable insights for investors, traders, and financial analysts.*

## II. Methodology

1. *Data Collection and Preprocessing:*
    - *The dataset is obtained from the* ***National Stock Exchange of India (NSEI)*** *official website for completeness and correctness.*
    - *Preprocessing involves handling missing values, scaling features, and ensuring data integrity and understanding the data.*
2. *Exploratory Data Analysis:*
    - *The data is organized into various chunks showcasing different parts of data making it further easy to analysis using Python Libraries like Numpy and Pandas.*
3. *Visualization of Results:*
    - *Various graphs have been plotted to showcase the data graphically for better understanding of the NIFTY 50 dataset.*
    - *Interactive visualizations using Matplotlib, Seaborn, enhance engagement and understanding of the dataset.*

# III. Documentation

*The project code is well-documented with clear explanations of each step. Annotations provide insights into decision-making processes. External libraries and resources are appropriately credited in the documentation.*

## ***Data Preprocessing:***

*Import necessary libraries —*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import warnings
warnings.filterwarnings('ignore')
```

*Reading Data from CSV file and displaying first few rows —*

```
df = pd.read_csv('Stock_DF.csv')
df.head()
```

*Output —*

| | Date | Open | High | Low | Close | Shares Traded | Turnover (₹ Cr) |
|---|---|---|---|---|---|---|---|
| 0 | 06-FEB-2023 | 17818.55 | 17823.70 | 17698.35 | 17764.60 | 282544790 | 21864.88 |
| 1 | 07-FEB-2023 | 17790.10 | 17811.15 | 17652.55 | 17721.50 | 354395693 | 23611.08 |
| 2 | 08-FEB-2023 | 17750.30 | 17898.70 | 17744.15 | 17871.70 | 290994265 | 23733.40 |
| 3 | 09-FEB-2023 | 17885.50 | 17916.90 | 17779.80 | 17893.45 | 260854055 | 21529.97 |
| 4 | 10-FEB-2023 | 17847.55 | 17876.95 | 17801.00 | 17856.50 | 231991834 | 17063.99 |

*Checking if dataset has null values —*

```
df.isnull().sum()
```

*Output —*

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

*Calculating the total time duration for which we are carrying out the analysis —*

```
df['Date '].max()-df['Date '].min()
```

*Output —*

```
Timedelta('365 days 00:00:00')
```

*There are approximately 252 trading days in a year with an average of 21 days per month, or 63 days per quarter. Out of a possible 365 days, 104 days are weekends (Saturday and Sunday) when the stock exchanges are closed.*

*Next, the describe() function of Pandas is used to get high-level overview of how the stock performed in about last couple of months —*

```
df.iloc[-90:].describe().astype(int)
```

*Output —*

| | Open | High | Low | Close | Shares Traded | Turnover (₹ Cr) |
|---|---|---|---|---|---|---|
| count | 90 | 90 | 90 | 90 | 90 | 90 |
| mean | 20522 | 20598 | 20429 | 20517 | 268518831 | 25183 |
| std | 1009 | 1026 | 991 | 1011 | 91157880 | 9145 |
| min | 18928 | 19041 | 18837 | 18857 | 37311902 | 2482 |
| 25% | 19591 | 19684 | 19526 | 19628 | 200097171 | 18620 |
| 50% | 20398 | 20497 | 20345 | 20477 | 245231118 | 23537 |
| 75% | 21539 | 21676 | 21448 | 21520 | 333943243 | 30792 |
| max | 22080 | 22126 | 21969 | 22097 | 508761756 | 50114 |

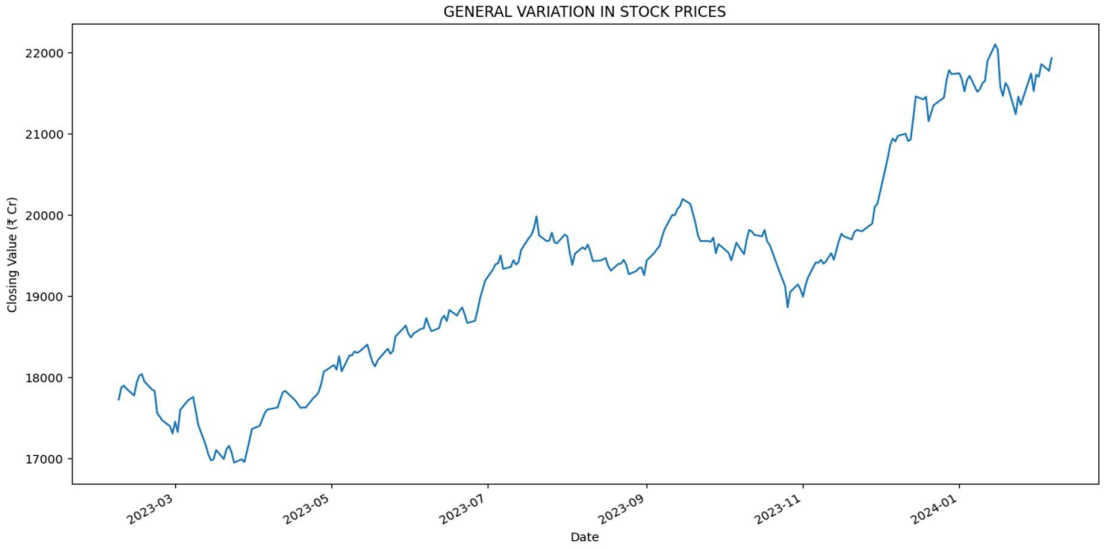
*In last 90 days, the average closing price for stock was about ₹20,517 Cr. For about 75% of time the stock was trading below ₹21,520 Cr and it clocked maximum of ₹21,969 Cr. The maximum volume of shares traded on a single day was 508761756 with median quantity being 245231118.*

## ***General Variation in Stock Prices:***

*Plotting the closing price of the stock over the period of a year to get a general idea of how the stock performed in the given period.*

```
df['Close '].plot(figsize=(15,8))
plt.ylabel('Closing Value (₹ Cr)')
plt.xlabel('Date')
plt.title("GENERAL VARIATION IN STOCK PRICES")
plt.show()
```

*Output —*



## ***Day-To-Day Percentage Change (Daily Returns):***

*Daily percentage change in the price of the stock is calculated on the basis of percentage change between 2 consecutive days' closing prices.*

*Accordingly, a new column 'Day_Perc_Change' is introduced denoting the daily returns in the price of the stock. This can be done using in-built pct_chang e() function in python.*

```
df['Day_Perc_Change'] = df['Close '].pct_change()*100
df.head()
```

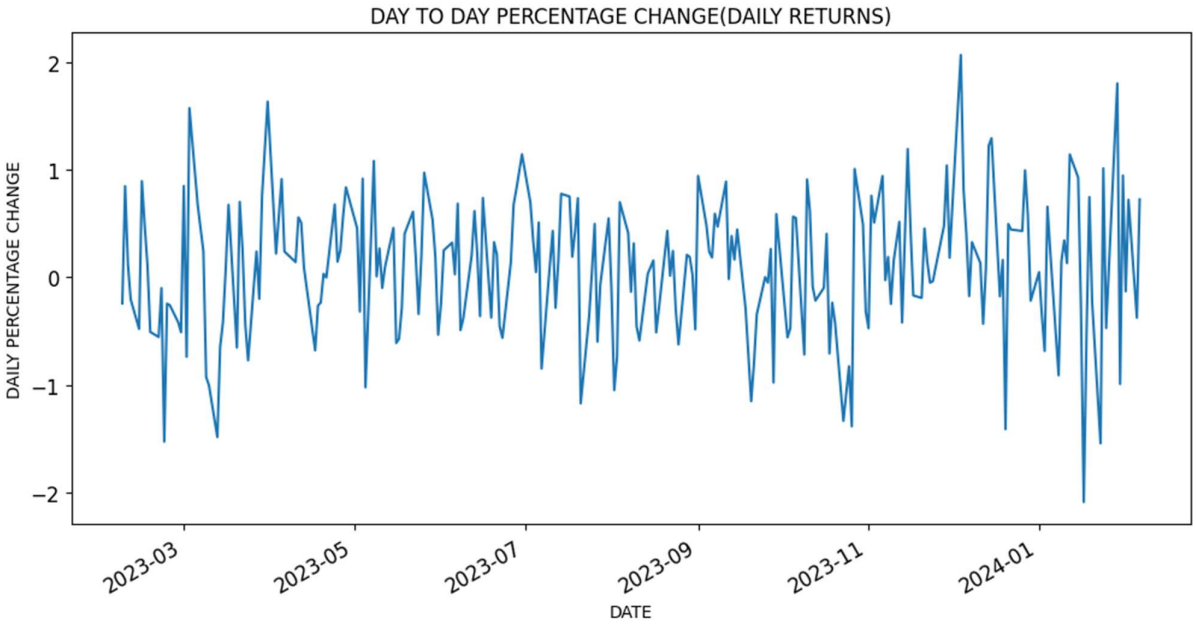
*Output —*

| Date | Date | Open | High | Low | Close | Shares Traded | Turnover (₹ Cr) | Day_Perc_Change |
|---|---|---|---|---|---|---|---|---|
| **2023-02-06** | 2023-02-06 | 17818.55 | 17823.70 | 17698.35 | 17764.60 | 282544790 | 21864.88 | NaN |
| **2023-02-07** | 2023-02-07 | 17790.10 | 17811.15 | 17652.55 | 17721.50 | 354395693 | 23611.08 | -0.242617 |
| **2023-02-08** | 2023-02-08 | 17750.30 | 17898.70 | 17744.15 | 17871.70 | 290994265 | 23733.40 | 0.847558 |
| **2023-02-09** | 2023-02-09 | 17885.50 | 17916.90 | 17779.80 | 17893.45 | 260854055 | 21529.97 | 0.121701 |
| **2023-02-10** | 2023-02-10 | 17847.55 | 17876.95 | 17801.00 | 17856.50 | 231991834 | 17063.99 | -0.206500 |

*Representing daily returns in form of a plot —*

```
df['Day_Perc_Change'].plot(figsize=(12,6),fontsize=12,title="DAY TO DAY PERCENTAGE CHANGE(DAILY RETURNS)", xlabel='DATE',ylabel='DAILY PERCENTAGE CHANGE')
```

*Output —*



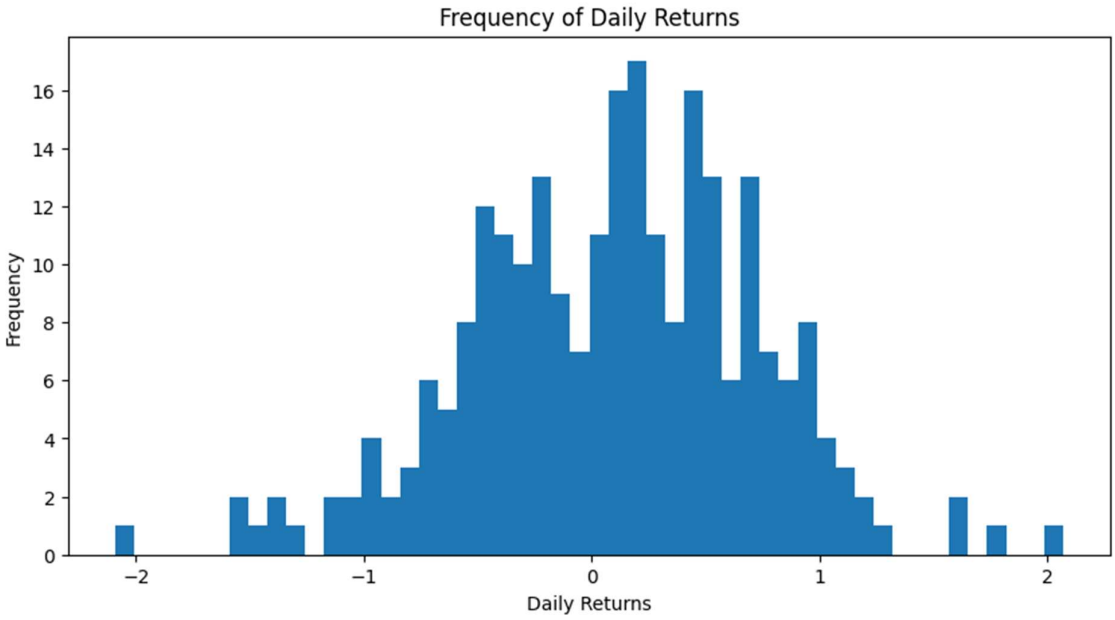

*It can be observed that for most of the days, the returns are between -1% to 1% with few spikes in between crossing 2% mark on both the sides.*

*Plotting daily returns distribution histogram —*

```
df['Day_Perc_Change'].hist(bins = 50, figsize = (10,5),grid=False)
plt.xlabel ('Daily Returns')
plt.ylabel('Frequency')
plt.title("Frequency of Daily Returns")
plt.show()

df.Day_Perc_Change.describe()
```

*Output —*




```
count    247.000000
mean       0.087317
std        0.635148
min       -2.089432
25%       -0.343063
50%        0.143807
75%        0.509059
max        2.066815
Name: Day_Perc_Change, dtype: float64
```

*The daily returns histogram is centered about origin. For the past year, the mean daily returns have been about 0.087 and for most of the days the daily return was less than 1% implying that the stock has been less volatile over the period. During the period, the highest % change in positive direction was observed to be 2.066% and was 2.089% in negative direction. Clearly, they didn't have any instances of 'bull run' or 'bear drop'.*

## ***Trend Analysis:***

A new column 'Trend' is added whose values are based on the day-to-day percentage change. Trend is determined from below relationship —

| DAILY PERCENTAGE CHANGE | TREND |
|---|---|
| -0.5 to 0.5 | Slight or No change |
| 0.5 to 1 | Slight Positive |
| -0.5 to -1 | Slight Negative |
| 1 to 3 | Positive |
| -1 to -3 | Negative |
| 3 to 7 | Among top gainers |
| -3 to -7 | Among top losers |
| Above 7 | Bull run |
| Below -7 | Bear drop |

```
def trend(x):
  if x > -0.5 and x <= 0.5:
    return 'Slight or No change'
  elif x > 0.5 and x <= 1:
    return 'Slight Positive'
  elif x > -1 and x <= -0.5:
    return 'Slight Negative'
  elif x > 1 and x <= 3:
    return 'Positive'
  elif x > -3 and x <= -1:
    return 'Negative'
  elif x > 3 and x <= 7:
    return 'Among top gainers'
  elif x > -7 and x <= -3:
    return 'Among top losers'
  elif x > 7:
    return 'Bull run'
  elif x <= -7:
    return 'Bear drop'
df['Trend']= np.zeros(df['Day_Perc_Change'].count())
df['Trend']= df['Day_Perc_Change'].apply(lambda x:trend(x))
df.head()
```

*Output —*

| Date | Date | Open | High | Low | Close | Shares Traded | Turnover (₹ Cr) | Day_Perc_Change | Trend |
|---|---|---|---|---|---|---|---|---|---|
| **2023-02-07** | 2023-02-07 | 17790.10 | 17811.15 | 17652.55 | 17721.50 | 354395693 | 23611.08 | -0.242617 | Slight or No change |
| **2023-02-08** | 2023-02-08 | 17750.30 | 17898.70 | 17744.15 | 17871.70 | 290994265 | 23733.40 | 0.847558 | Slight Positive |
| **2023-02-09** | 2023-02-09 | 17885.50 | 17916.90 | 17779.80 | 17893.45 | 260854055 | 21529.97 | 0.121701 | Slight or No change |
| **2023-02-10** | 2023-02-10 | 17847.55 | 17876.95 | 17801.00 | 17856.50 | 231991834 | 17063.99 | -0.206500 | Slight or No change |
| **2023-02-13** | 2023-02-13 | 17859.10 | 17880.70 | 17719.75 | 17770.90 | 231276483 | 17406.31 | -0.479377 | Slight or No change |

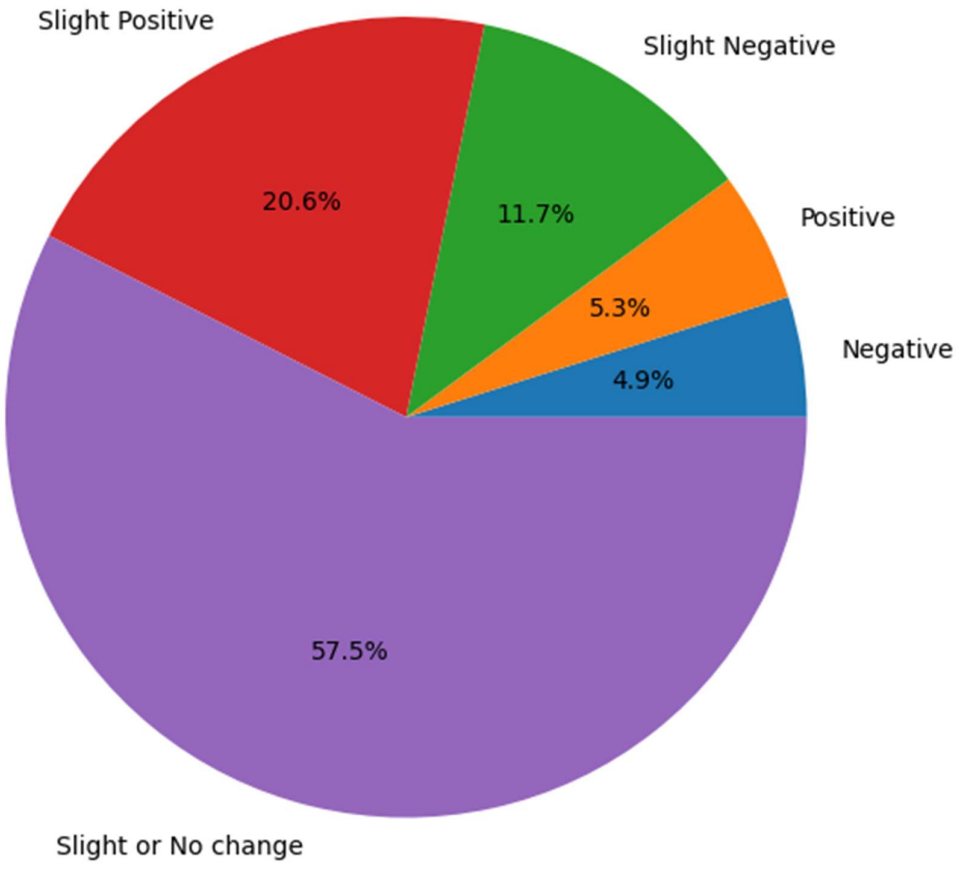
*How the stock was trending in past one year can be visualized as a pie chart, with each sector representing the percentage of days each trend occurred. A pie chart is plotted for the 'Trend' column to visualize the relative frequency of each trend category.*

*The groupby() function is used with the trend column to aggregate all days with the same trend into a single group before plotting the pie chart.*

*Visualizing Trend Frequency with Pie-Chart —*

```
DFC_pie_data = df.groupby('Trend')
pie_label = sorted([i for i in df.loc[:, 'Trend'].unique()])
plt.pie(DFC_pie_data['Trend'].count(), labels = pie_label,
        autopct = '%1.1f%%', radius = 1.5)
plt.show()
```

*Output —*



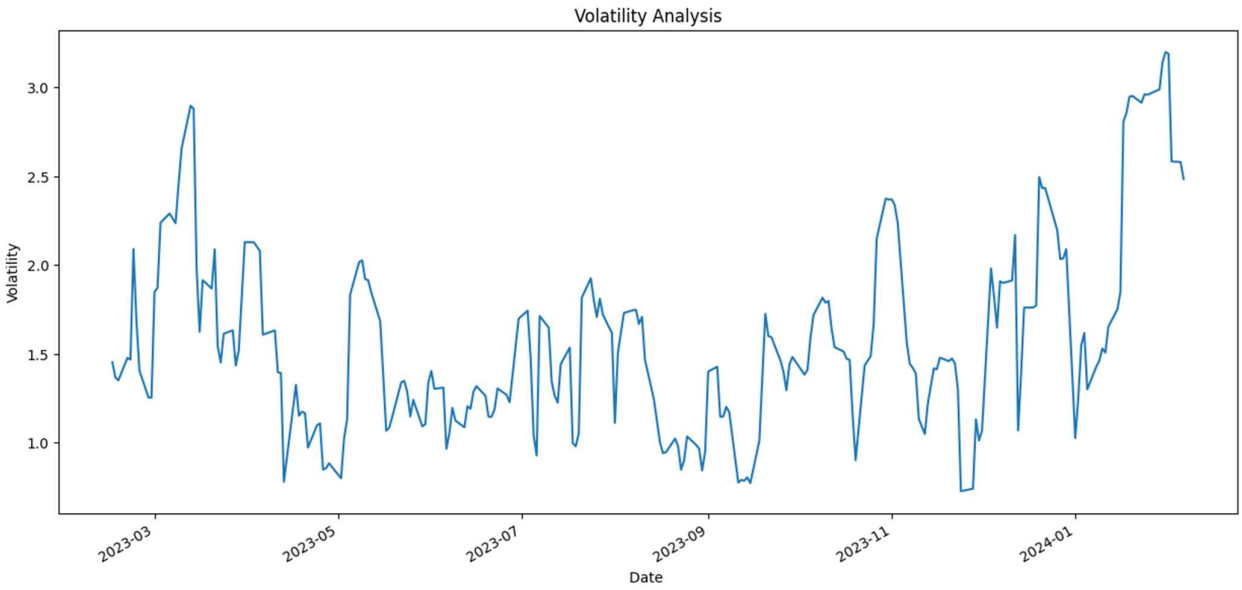
## ***Volatility Analysis:***

***Volatility*** *is one of the most important pillars in financial markets. A stock is said to have high volatility if its value can change dramatically within a short span of time. On other hand, lower volatility means that value of stock tends to be relatively steady over a period of time. These movements are due to several factors including demand and supply, sentiment, corporate actions, greed, and fear, etc. Mathematically, volatility is measured using a statistical measure called 'standard deviation', which measures an asset's departure from its average value.*

*The intraday returns (daily returns) of the stock are calculated. Now, the 7-day rolling mean (also called moving average) of the daily returns and the standard deviation (which is square root of the variance) is calculated using pandas 'rolling()' function and 'std()' function and the values are plotted.*

```
vol = df["Day_Perc_Change"].rolling(7).std()*np.sqrt(7)
vol.plot(figsize = (15, 7))
plt.ylabel ('Volatility')
plt.title("Volatility Analysis")
```

*Output —*




*Many traders and investors seek out higher volatility investments in order to make higher profits. If a stock does not move, not only it has low volatility, but also it has low gain potential. On the other hand, a stock or other security with a very high volatility level can have tremendous profit potential, but the risk is equally high.*

# IV. Conclusion

*In this project, we embarked on a comprehensive journey to analyse and visualize the NIFTY 50 dataset spanning from February 6th, 2023, to February 6th, 2024. Our analysis encompassed several crucial steps, including data preprocessing, calculating the general variation of the stock, computing daily returns, and conducting trend and volatility analyses.*

*During the data preprocessing phase, we meticulously cleaned and prepared the dataset, ensuring accuracy and consistency in our subsequent analyses. By handling missing values, removing duplicates, and standardizing data formats, we laid a robust foundation for our analysis.*

*Next, we delved into calculating the general variation of the NIFTY 50 stock, providing insights into the overall movement and volatility of the index during the specified timeframe. This analysis allowed us to gauge the level of market fluctuations and understand the inherent risks associated with investing in the NIFTY 50 index.*

*Subsequently, we computed daily returns, a fundamental metric for assessing the performance of financial instruments over time. By quantifying the percentage change in the NIFTY 50 index on a daily basis, we gained valuable insights into the index's day-to-day fluctuations and potential trends.*

*Our project also involved plotting trend analysis, enabling us to visualize the historical performance of the NIFTY 50 stock over the designated one-year period. Through line plots and candlestick charts, we identified patterns, trends, and potential areas of interest for further exploration.*

*Furthermore, we conducted volatility analysis, examining the variability and dispersion of returns in the NIFTY 50 dataset. By employing statistical measures such as standard deviation and variance, we quantified the level of volatility exhibited by the index, providing investors with crucial information for risk management and portfolio optimization.*

*In **conclusion**, our data analysis and visualization efforts have provided stakeholders with valuable insights into the dynamics of the NIFTY 50 stock from February 6th, 2023, to February 6th, 2024. By leveraging Python's powerful libraries and techniques, we have equipped investors, traders, and financial analysts with actionable insights to make informed decisions in the ever-evolving landscape of financial markets. Moving forward, further exploration and refinement of our analysis techniques could offer even deeper insights into the behaviour and trends of the NIFTY 50 index, facilitating more informed decision-making in the realm of investments.*

# V. References:

- ***Collab Notebook link:*** *-*

*https://colab.research.google.com/drive/1goWZEnX18r_U_PgIj7QD4-JM3FjHHmkP?usp=sharing*

- ***NIFTY 50 Dataset CSV File*** *is directly downloaded from official* ***National Stock Exchange of India (NSEI)*** *website for the time span of a year* ***(06/02/2023 to 06/02/2024)***