Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Report on Mini Project

Time Series Analysis (DJ19DSC5012)
AY: 2021-22

Will Bill open the gates of Microsoft?

Jay Bhanushali (60009200047)
Pushkar Waykole(60009200032)
Mannan Singhvi(60009200045)

Guided By
Prof. Pradnya Saval

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Time-series analysis is a method of analyzing a collection of data points over a period of time. Instead of recording data points intermittently or randomly, time series analysts record data points at consistent intervals over a set period of time. While time-series data is information gathered over time, various types of information describe how and when that information was gathered.

Our aim is to study the Closing price of Microsoft Dataset and forecast the closing price for future.

CHAPTER 2. DATA DESCRIPTION

The link to the dataset we used is:

https://www.kaggle.com/datasets/varpit94/microsoft-stock-data

Our dataset has 5 columns

	Date	Open	High	Low	Close	Volume
0	04-01-2015	40.60	40.76	40.31	40.72	36865322
1	04-02-2015	40.66	40.74	40.12	40.29	37487476
2	04-06-2015	40.34	41.78	40.18	41.55	39223692
3	04-07-2015	41.61	41.91	41.31	41.53	28809375
4	04-08-2015	41.48	41.69	41.04	41.42	24753438

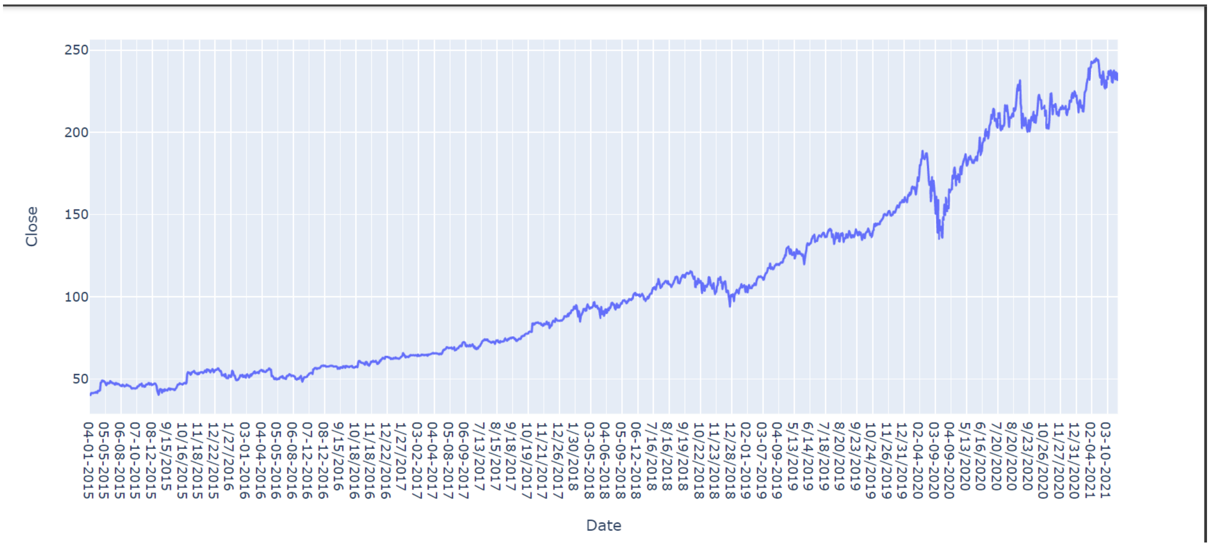
```
[4]  1  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1511 entries, 0 to 1510
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    1511 non-null   object
 1   Open    1511 non-null   float64
 2   High    1511 non-null   float64
 3   Low     1511 non-null   float64
 4   Close   1511 non-null   float64
 5   Volume  1511 non-null   int64
dtypes: float64(4), int64(1), object(1)
memory usage: 71.0+ KB
```

The date column is of type object and the Open,High,Low,Close are of type float 64.

The Volume is of type int 64.

The Line chart of Closing price vs Date is:



We can see that there is an uptrend in the time series data.

Also the mean is not constant over time ,hence our data is not stationary.

CHAPTER 3. OBJECTIVE

The objective of our mini project is to study the trend and properties of the Microsoft Stock's Closing price and correctly predict the closing price of the stock in future.

CHAPTER 4. DATA CLEANING

Checking the null values

```
df.isna().sum()
```

```
Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
dtype: int64
```

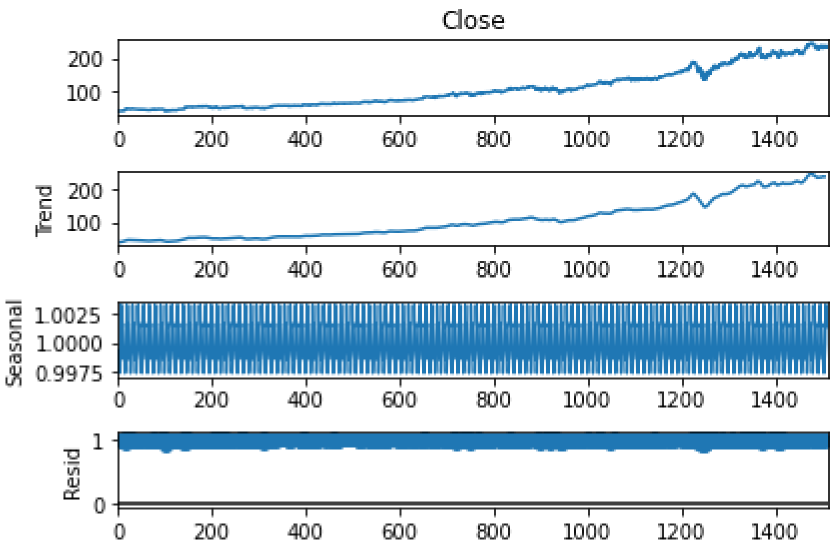
There are no null values in our dataset.

CHAPTER 5. DATA DECOMPOSITION

Decomposition is a statistical task in which the Time Series data is decomposed into several components or extracting seasonality, trend from a series data. These components are defined as follows:

- Level: The average value in the series.
- Trend: The increasing or decreasing value in the series.
- Seasonality: The repeating short-term cycle in the series.
- Noise: The random variation in the series.

```
result=seasonal_decompose(df['Close'], model='multiplicable', period=12)
result.plot()
```



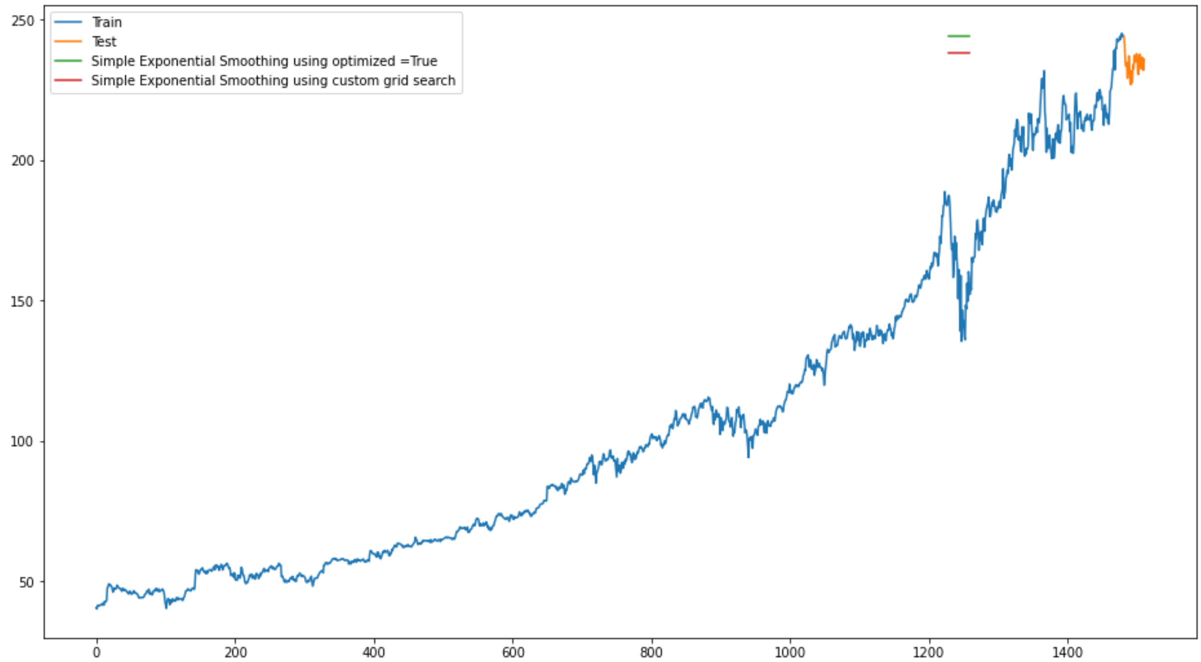
CHAPTER 6. SMOOTHING METHODS

Simple Exponential Smoothing is used for time series prediction when the data particularly does not follow any:

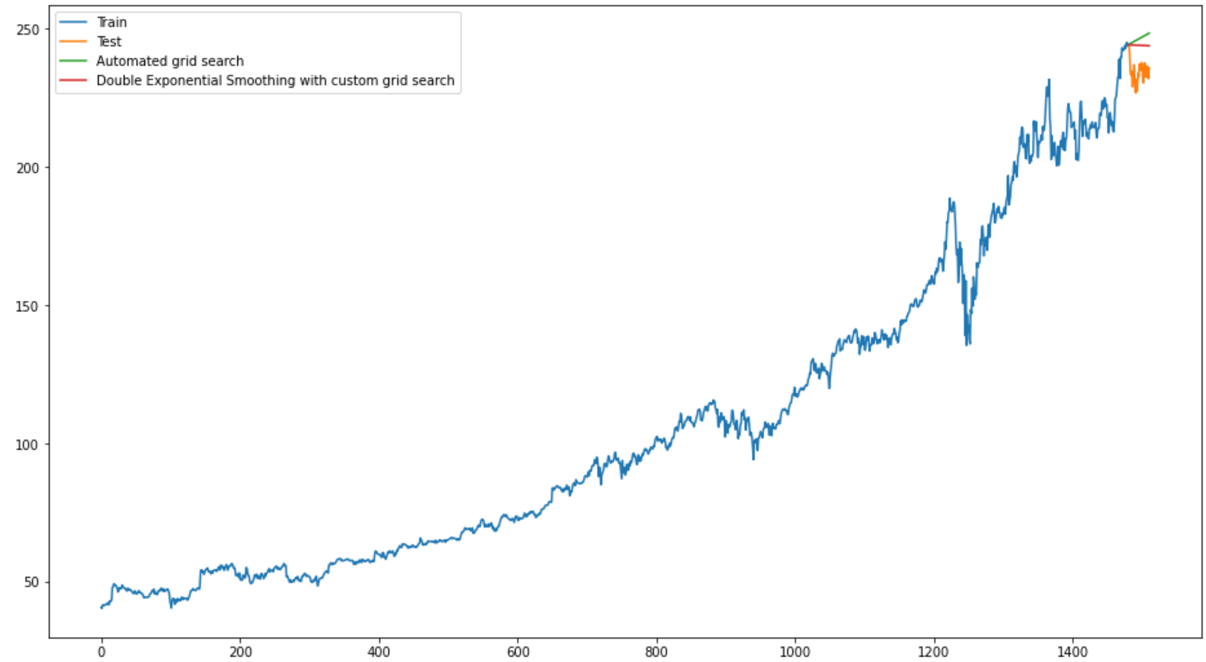
1. Trend: An upward or downward slope
2. Seasonality: Shows a particular pattern due to seasonal factors like Hours, days, Year, etc.

But our data has uptrend so we cannot use simple and double exponential smoothing methods.

Result of Simple Smoothing



Result of Double smoothing


Train
Test
Automated grid search
Double Exponential Smoothing with custom grid search
250
200
150
100
50
0
200
400
600
800
1000
1200
1400

CHAPTER 7. TESTING STATIONARY

A time series is said to be "stationary" if it has no trend, exhibits constant variance over time, and has a constant autocorrelation structure over time.

One way to test whether a time series is stationary is to perform an augmented Dickey-Fuller test, which uses the following null and alternative hypotheses:

H0: The time series is non-stationary. In other words, it has some time-dependent structure and does not have constant variance over time.

HA: The time series is stationary.

If the p-value from the test is less than some significance level (e.g. α = .05), then we can reject the null hypothesis and conclude that the time series is stationary.

```
Augmented_Dickey_Fuller_Test_func(df['Close'] , 'Close')
```

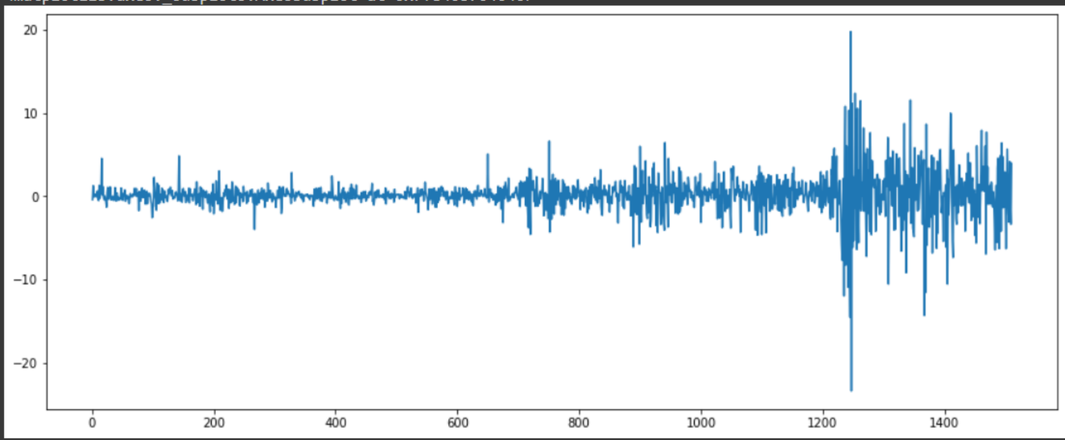
```
Results of Dickey-Fuller Test for column: Close
Test Statistic                                                  1.737136
p-value                                                         0.998216
No Lags Used                                                          24
Number of Observations Used                                         1486
 Critical values           {'1%': -3.4347582315402434, '5%': -2.863486949...
Critical Value (1%)                                            -3.434758
Critical Value (5%)                                            -2.863487
Critical Value (10%)                                           -2.567807
dtype: object
Conclusion:====>
Fail to reject the null hypothesis
Data is non-stationary
```

The data is not stationary hence we need to apply first differencing

1st Differencing

```
[43] 1 close = df['Close'].diff().dropna()
     2 close.plot(figsize=(15,6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f846b764040>
```



After applying the first differencing the data became stationary

```
[44] 1 Augmented_Dickey_Fuller_Test_func(close , 'Close')
```

```
Results of Dickey-Fuller Test for column: Close
Test Statistic                                                  -10.038331
p-value                                                                0.0
No Lags Used                                                            24
Number of Observations Used                                           1485
 Critical values              {'1%': -3.43476120520139, '5%': -2.86348826217...
Critical Value (1%)                                              -3.434761
Critical Value (5%)                                              -2.863488
Critical Value (10%)                                             -2.567807
dtype: object
Conclusion:====>
Reject the null hypothesis
Data is stationary
```

CHAPTER 8. JUSTIFICATION WHY IT IS A TIME SERIES PROBLEM.

As stock prices are updated every second the analysis of Microsoft Stock price is a time series problem

CHAPTER 9. IMPLEMENTATION AND INTERPRETATION FOR FORECAST

For forecasting we can use several methods like:

1.ARIMA

2GARCH

3.CNN

1. ARIMA

```
from pmdarima import auto_arima
arima = auto_arima(df['Close'], trace=True, suppress_warnings=True)
arima.summary()
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=6761.238, Time=1.29 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=6876.786, Time=0.06 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=6760.385, Time=0.13 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=6770.738, Time=0.62 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=6879.320, Time=0.16 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=6762.379, Time=0.47 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=6762.380, Time=0.47 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=6764.095, Time=1.19 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=6766.263, Time=0.09 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0] intercept
Total fit time: 4.497 seconds
```

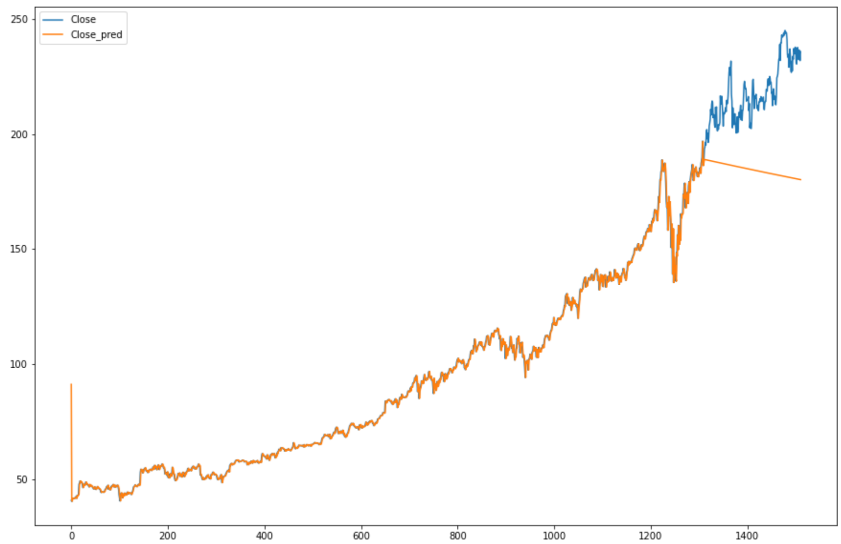
SARIMAX Results

Dep. Variable:	y	**No. Observations:**	1511
Model:	SARIMAX(1, 1, 0)	**Log Likelihood**	-3377.192
Date:	Thu, 08 Dec 2022	**AIC**	6760.385
Time:	15:11:08	**BIC**	6776.344
Sample:	0	**HQIC**	6766.328
	- 1511		
Covariance Type:	opg		

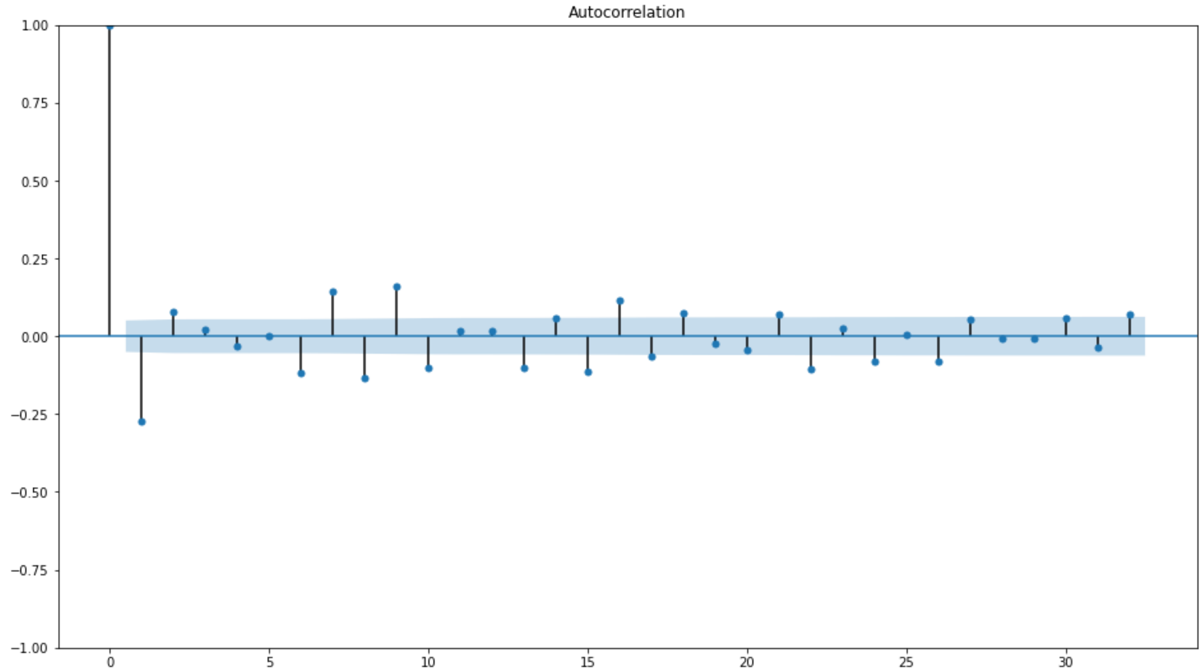
	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.1641	0.059	2.788	0.005	0.049	0.280
ar.L1	-0.2748	0.010	-28.723	0.000	-0.294	-0.256
sigma2	5.1303	0.074	69.006	0.000	4.985	5.276

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	9082.00
Prob(Q):	0.99	**Prob(JB):**	0.00
Heteroskedasticity (H):	20.43	**Skew:**	-0.51
Prob(H) (two-sided):	0.00	**Kurtosis:**	14.97

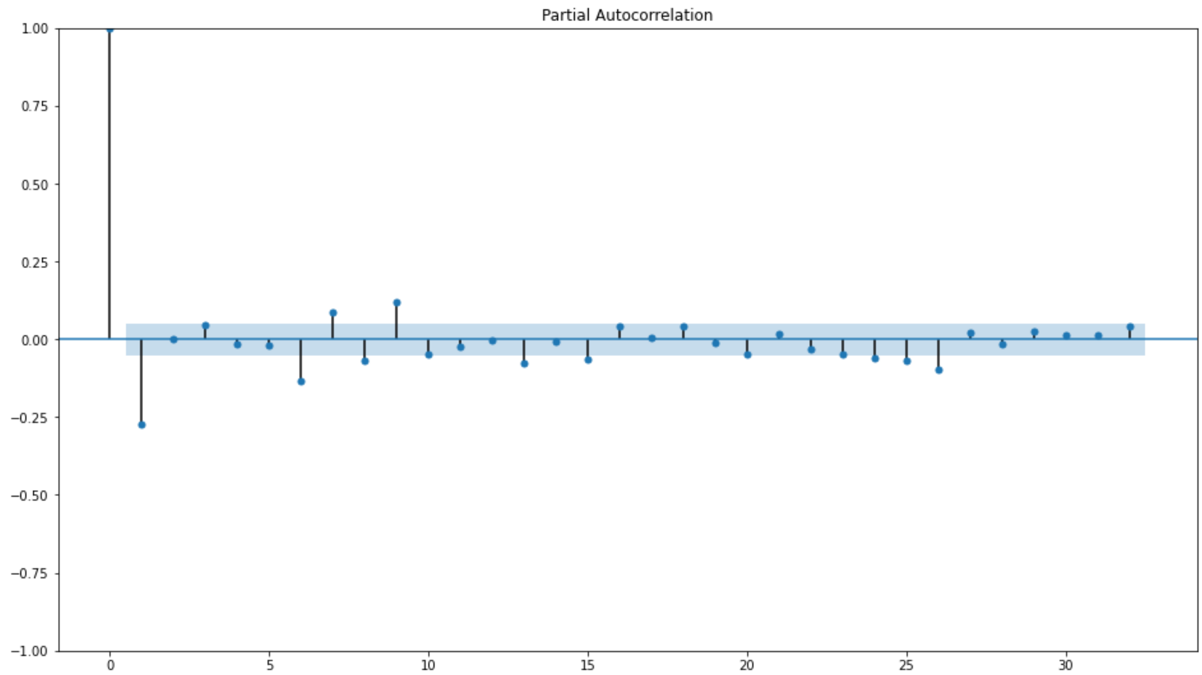
Prediction using ARIMA



The autocorrelation plot



The partial Autocorrelation plot



2.GARCH

We are fitting the GARCH(1,1) model as the order is observed from ACF and PACF plot.

```
from arch import arch_model
garch = arch_model(df,vol='Garch', p=1, q=1)
result = garch.fit()
result.summary()
```

```
Optimization terminated successfully    (Exit mode 0)
            Current function value: 2666.964488258338
            Iterations: 14
            Function evaluations: 81
            Gradient evaluations: 14
```

Constant Mean - GARCH Model Results

Dep. Variable:	Close	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GARCH	**Log-Likelihood:**	-2666.96
Distribution:	Normal	**AIC:**	5341.93
Method:	Maximum Likelihood	**BIC:**	5363.21
		No. Observations:	1510
Date:	Thu, Dec 08 2022	**Df Residuals:**	1509
Time:	15:11:18	**Df Model:**	1

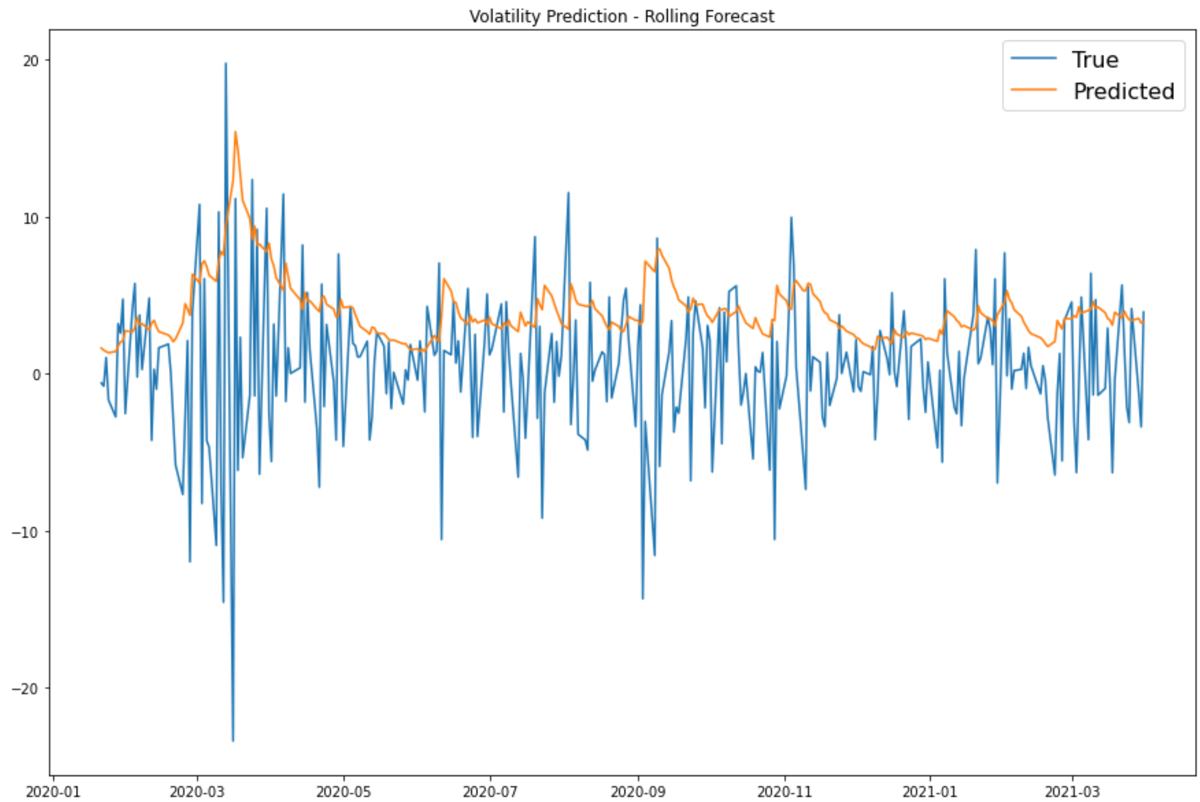
Mean Model

	coef	**std err**	**t**	**P>|t|**	**95.0% Conf. Int.**
mu	0.0861	2.619e-02	3.287	1.012e-03	[3.476e-02, 0.137]

Volatility Model

	coef	**std err**	**t**	**P>|t|**	**95.0% Conf. Int.**
omega	0.0585	3.038e-02	1.926	5.414e-02	[-1.040e-03, 0.118]
alpha[1]	0.1783	6.586e-02	2.707	6.790e-03	[4.920e-02, 0.307]
beta[1]	0.8147	5.943e-02	13.707	9.192e-43	[0.698, 0.931]

The prediction for test size=20%



The result is very close to the actual volatility of the data.

3.CNN

We have used the Relu function as the activation function and MSE as the loss function with Adam optimiser

The result of prediction

CHAPTER 10. REASONS FOR SELECTING THE TIME SERIES MODEL

Since our data comes under a financial category which has volatility present in it,the GARCH model gives more accurate predictions as compared to other methods.

CHAPTER 11. COMPARATIVE RESULT ANALYSIS

The GARCH and CNN model gives accurate predictions and can be used to predict the future closing prices of the stock.

CHAPTER 12. GOOGLE COLAB LINK

Microsoft Stock price analysis

CHAPTER 13. CONCLUSION

The GARCH model is used when there is volatility in the data.

And thus GARCH is the most suited model to use in financial data.

CHAPTER 14. FUTURE SCOPE

In the future we are planning to make a web app that can do all the analysis done above for various stocks and predict the future prices accurately.

CHAPTER 15. REFERENCES

[Data Smoothing]

[Decomposition in Time series]

[GARCH Model]