Comparison of Various Time Series Models

Group No 6

Problem Statement

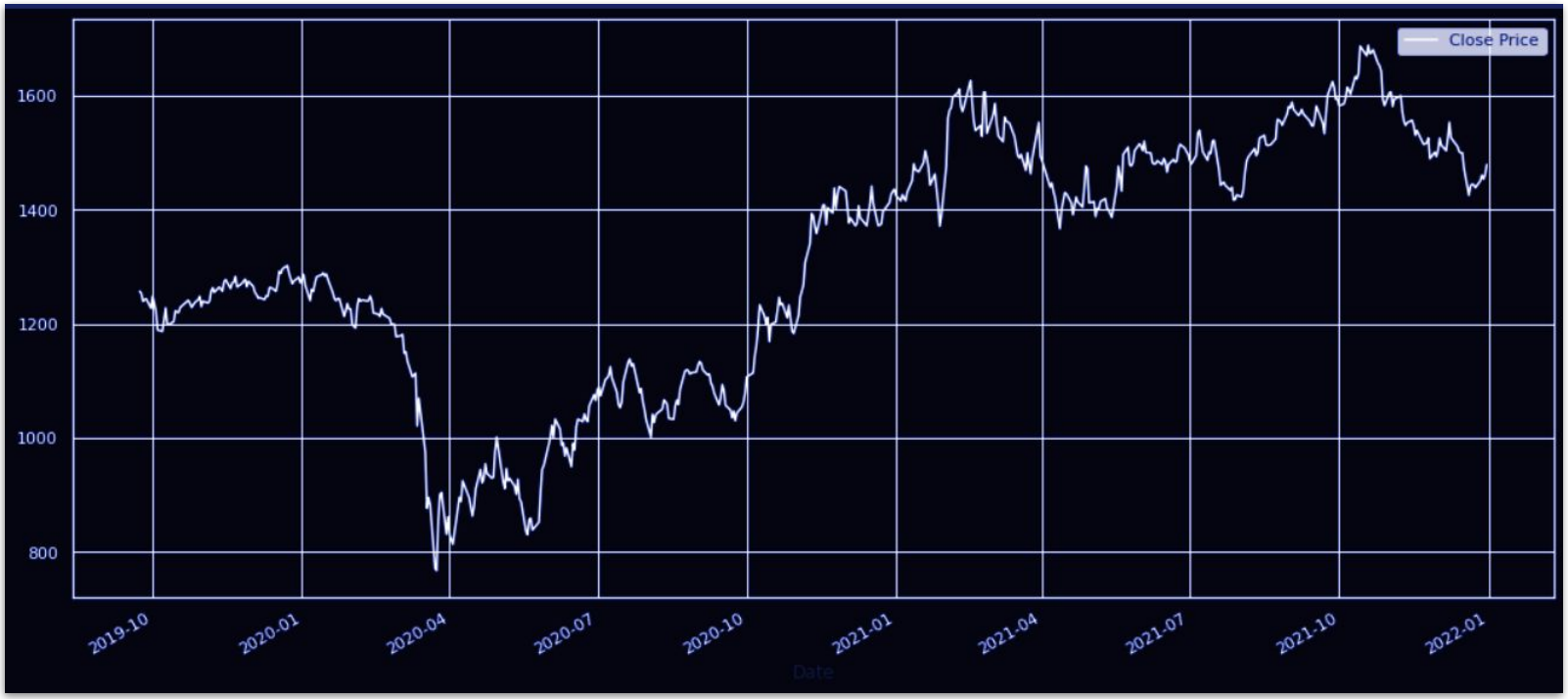
“Stock market is a complex and challenging system where people will either gain money or lose their entire life savings, We are trying to come up with the best model which has better accuracy for the HDFC Bank share price”

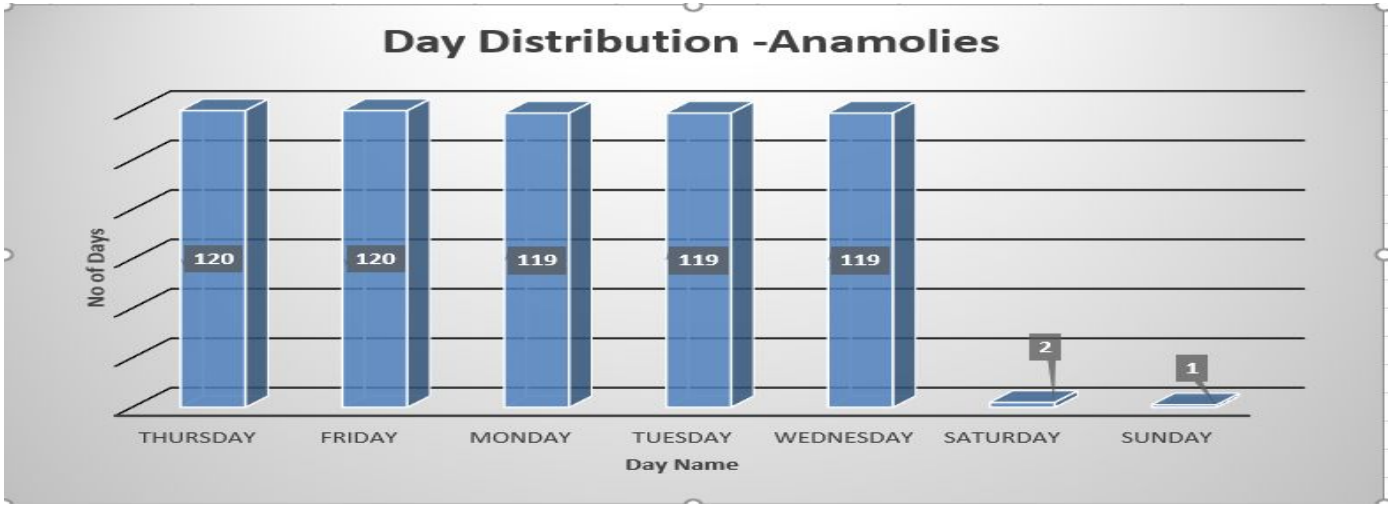
Data Dictionary

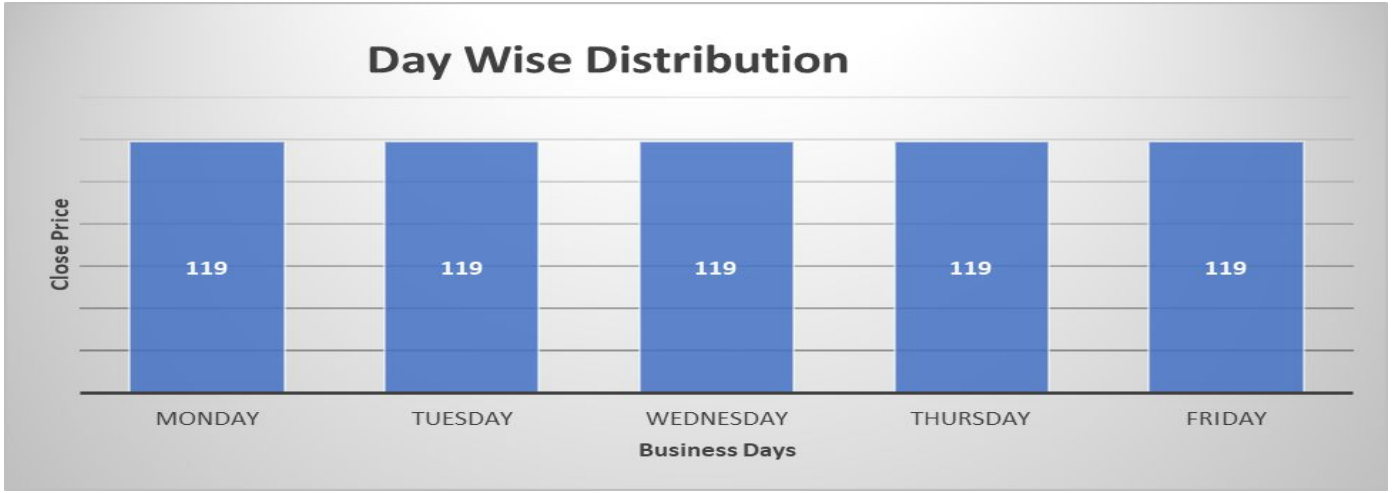
Variable	Refers	Datatype
Date	Date at which the price was noted	Datetime
Open Price	Current day's open price	float
Close Price	Closing Price of the stock	float
High Price	Current day's highest Price	float
Low Price	Current day's lowest price	float

Pre-Processing
&
Exploratory Data Analysis

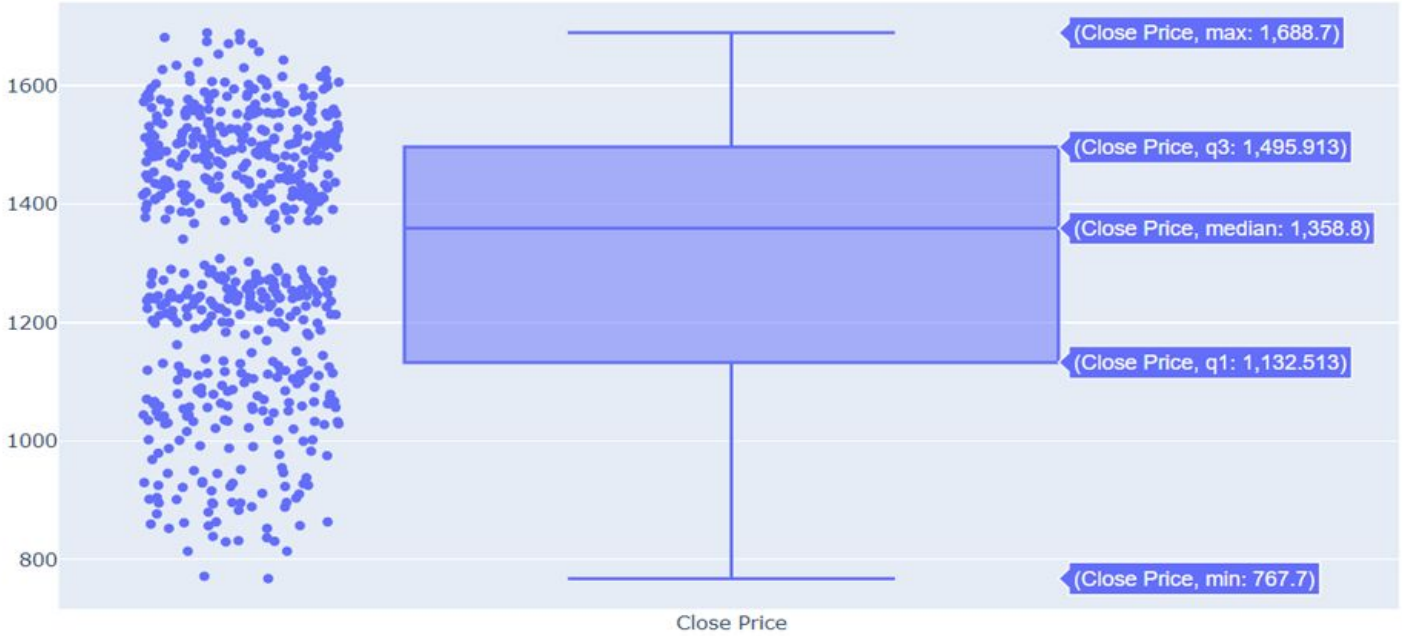
Data Distribution




Day Distribution -Anamolies
No of Days
120
120
119
119
119
2
1
THURSDAY
FRIDAY
MONDAY
TUESDAY
WEDNESDAY
SATURDAY
SUNDAY
Day Name

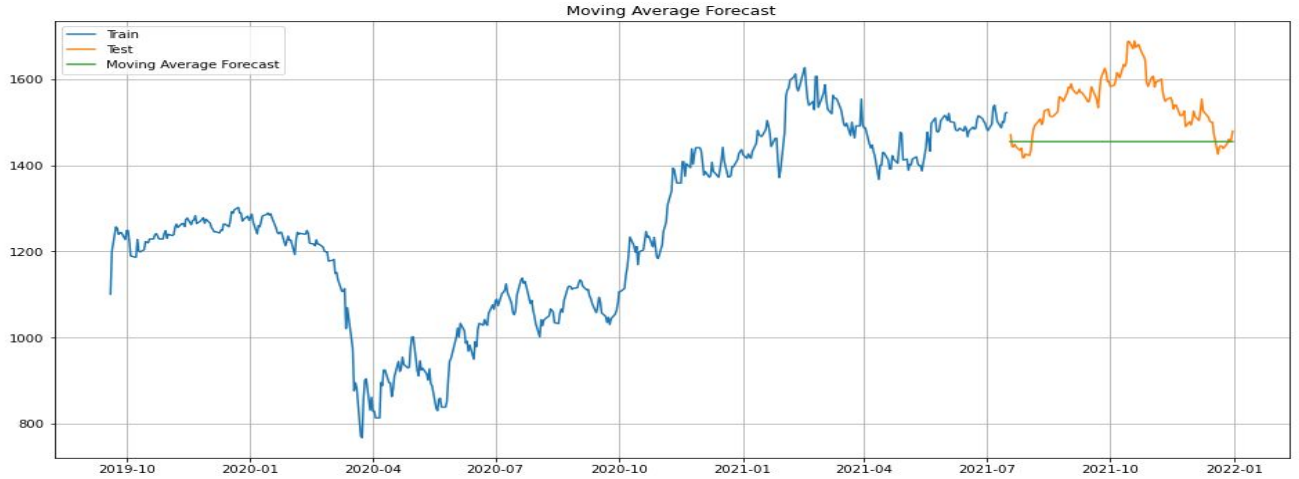

Day Wise Distribution
Close Price
119
119
119
119
119
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
Business Days

- 31 Missing Values found -Imputed using forward fill
- No Outliers Found

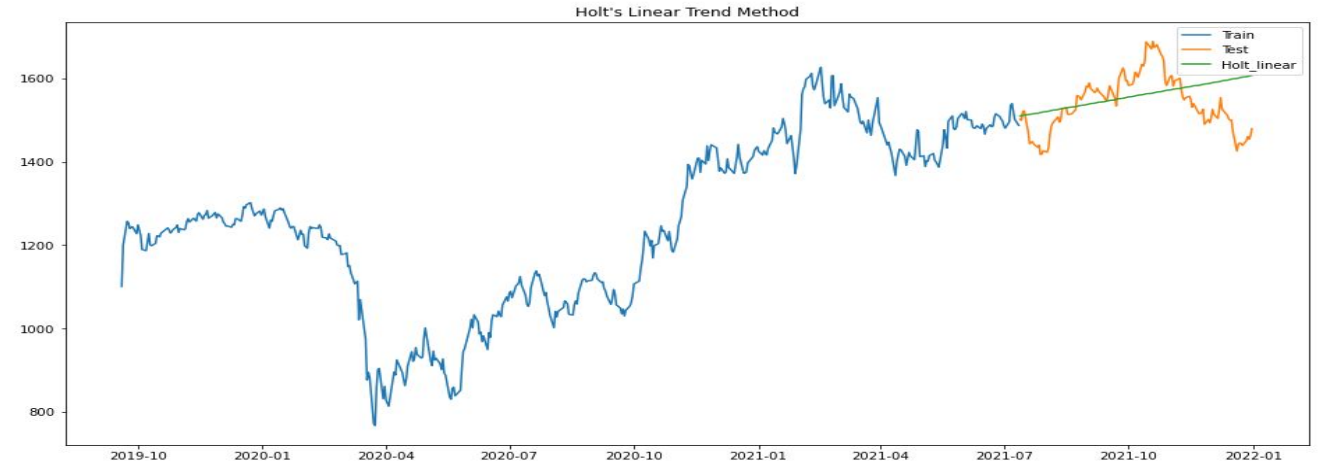


Base Models

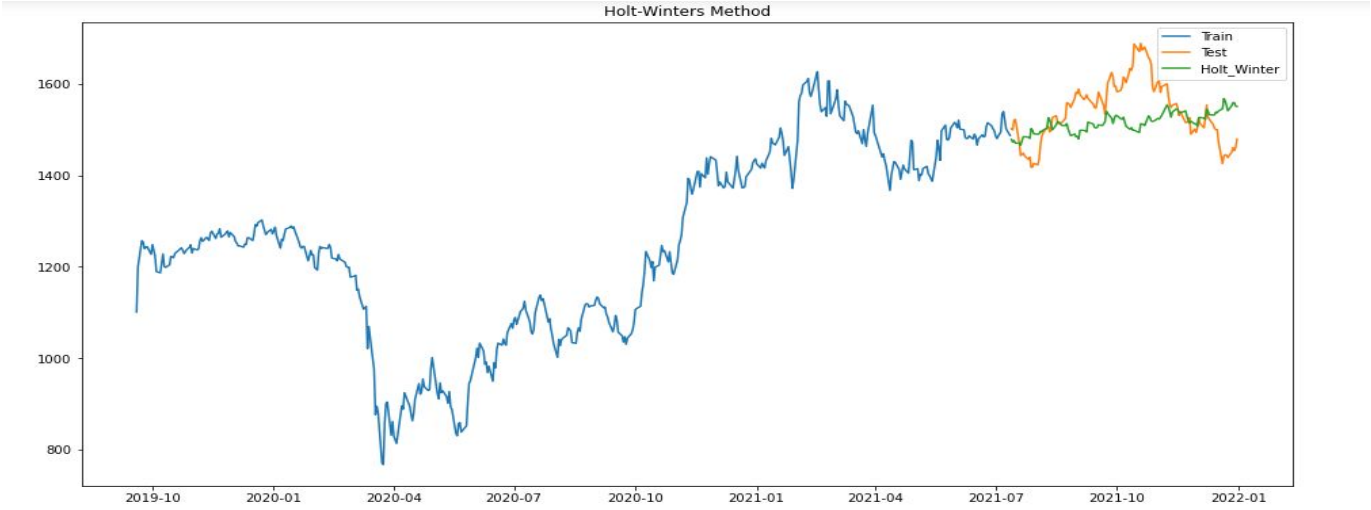
Moving Average



Holt's Linear Trend



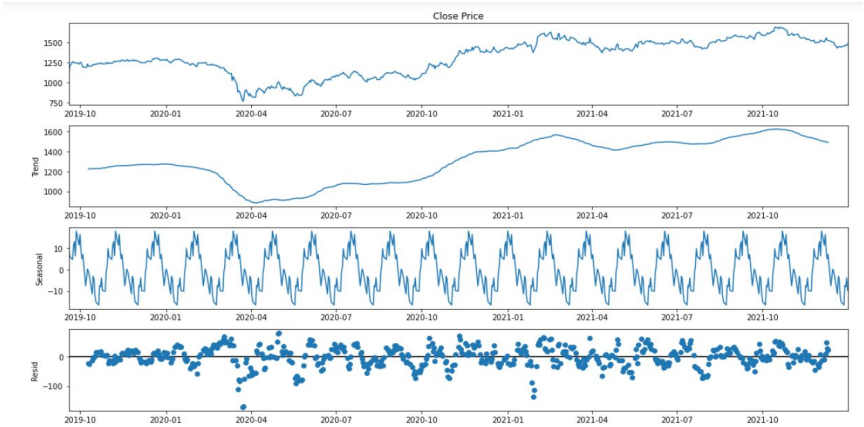
Holt's Winter Trend



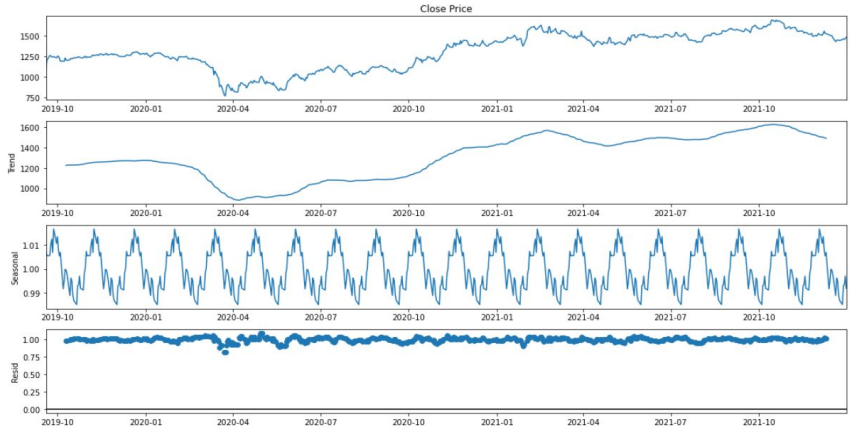
Performance Metrics

Model_Name	MSE	RMSE	MAE
Naive Approach	7114.858072	84.349618	68.891102
Simple Average	7114.858072	84.349618	68.891102
Moving Average	6726.594466	82.015818	66.632910
Simple Exponential Smoothing	6470.897086	80.441886	65.092771
Holt's Linear Trend method	5281.537399	72.674187	58.470691
Holt's Winter method	5472.769356	73.978168	58.359440

Additive Decomposition



Multiplicative Decomposition



Stationarity Check

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(df['Close Price'])
```

```
print(f'Test Statistics: {result[0]}')
print(f'p-value: {result[1]}')
print(f'Critial Values: {result[4]}')

if result[1] > 0.05 :
  print('Series is not Stationary')
else:
  print('Series is Stationary')
```

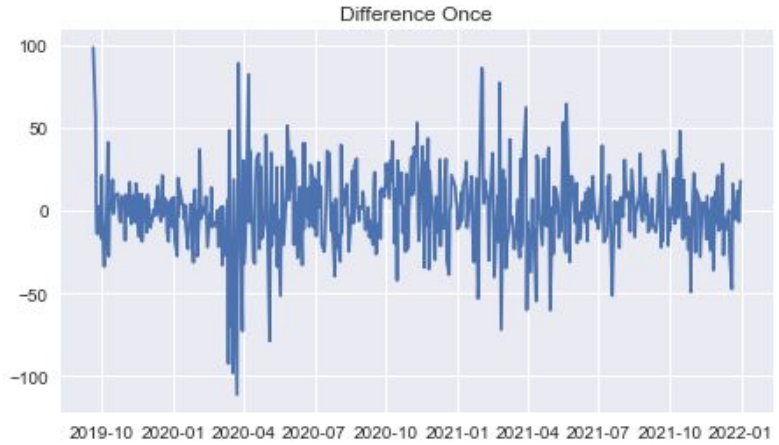
```
Test Statistics: -1.202703287847015
p-value: 0.672431745985324
Critial Values: {'1%': -3.442060292264578, '5%': -2.866705729876777, '10%': -2.569521171354946}
Series is not Stationary
```

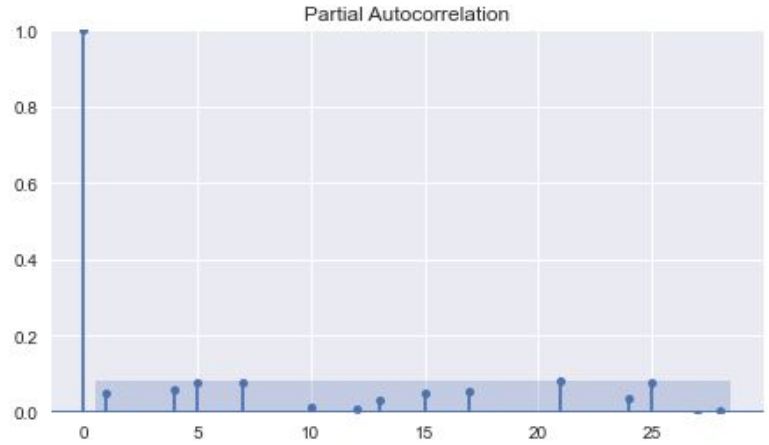
Differencing

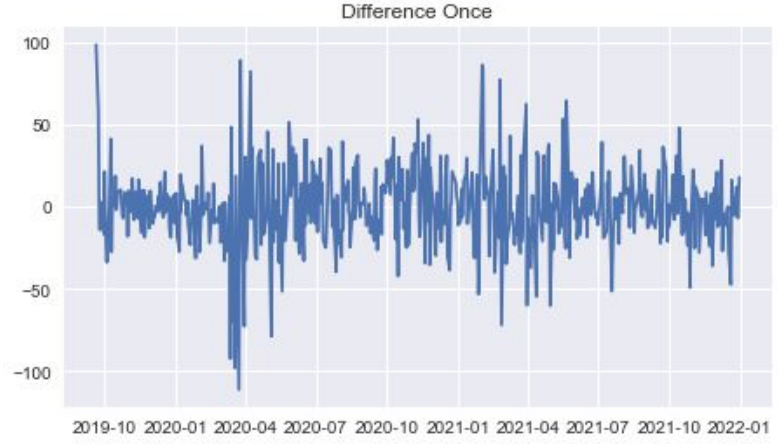
```
result = adfuller(df['Close Price'].diff().dropna())
```

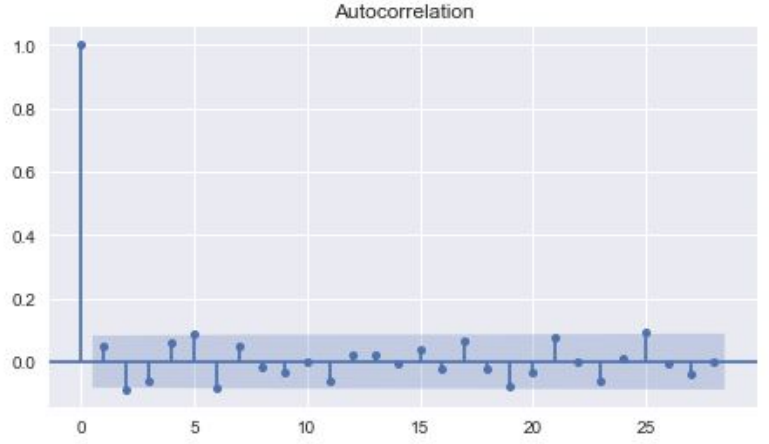
```
print(f'Test Statistics: {result[0]}')
print(f'p-value: {result[1]}')
print(f'Critial Values: {result[4]}')

if result[1] > 0.05 :
  print('Series is not Stationary')
else:
  print('Series is Stationary')
```

```
Test Statistics: -8.43651407556981
p-value: 1.8105665571755115e-13
Critial Values: {'1%': -3.442060292264578, '5%': -2.866705729876777, '10%': -2.569521171354946}
Series is Stationary
```


Difference Once
100
50
0
−50
−100
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
2021-10
2022-01


Partial Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25


Difference Once
100
50
0
−50
−100
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
2021-10
2022-01


Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25

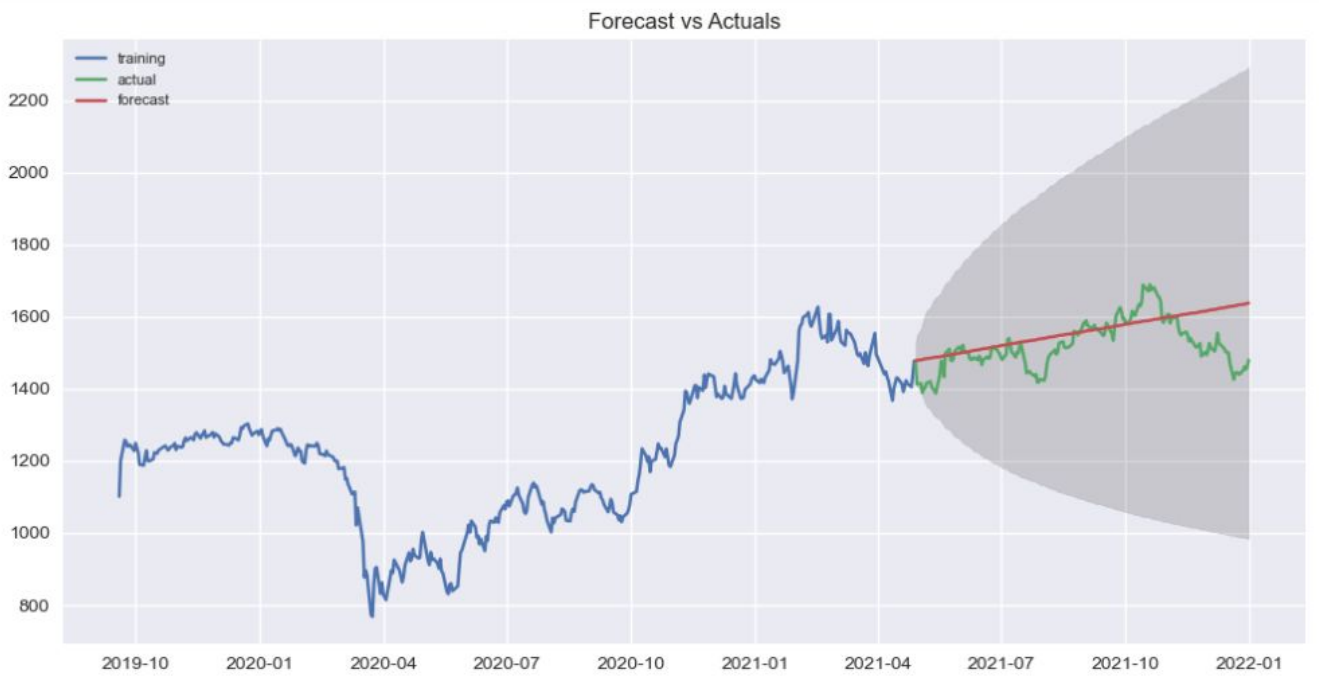
```
from pmdarima.arima.utils import ndiffs
y = df['Close Price']

## Adf Test
print(ndiffs(y, test='adf'))

# KPSS test
print(ndiffs(y, test='kpss'))

# PP test:
print(ndiffs(y, test='pp'))
```

```
1
1
1
```

ARIMA and SARIMA

ARIMA

	param	AIC
3	(0, 1, 0)	3714.448380
17	(2, 1, 2)	3715.195847
4	(0, 1, 1)	3715.614392
9	(1, 1, 0)	3715.702520
15	(2, 1, 0)	3716.432179



SARIMA

	param	seasonal	AIC
3119	(2, 1, 2)	(0, 1, 2, 12)	3380.753836
959	(0, 1, 2)	(0, 1, 2, 12)	3392.630875
1019	(0, 1, 2)	(1, 1, 2, 12)	3392.686622
2099	(1, 1, 2)	(1, 1, 2, 12)	3394.360341
2039	(1, 1, 2)	(0, 1, 2, 12)	3394.391278

Performance metrics

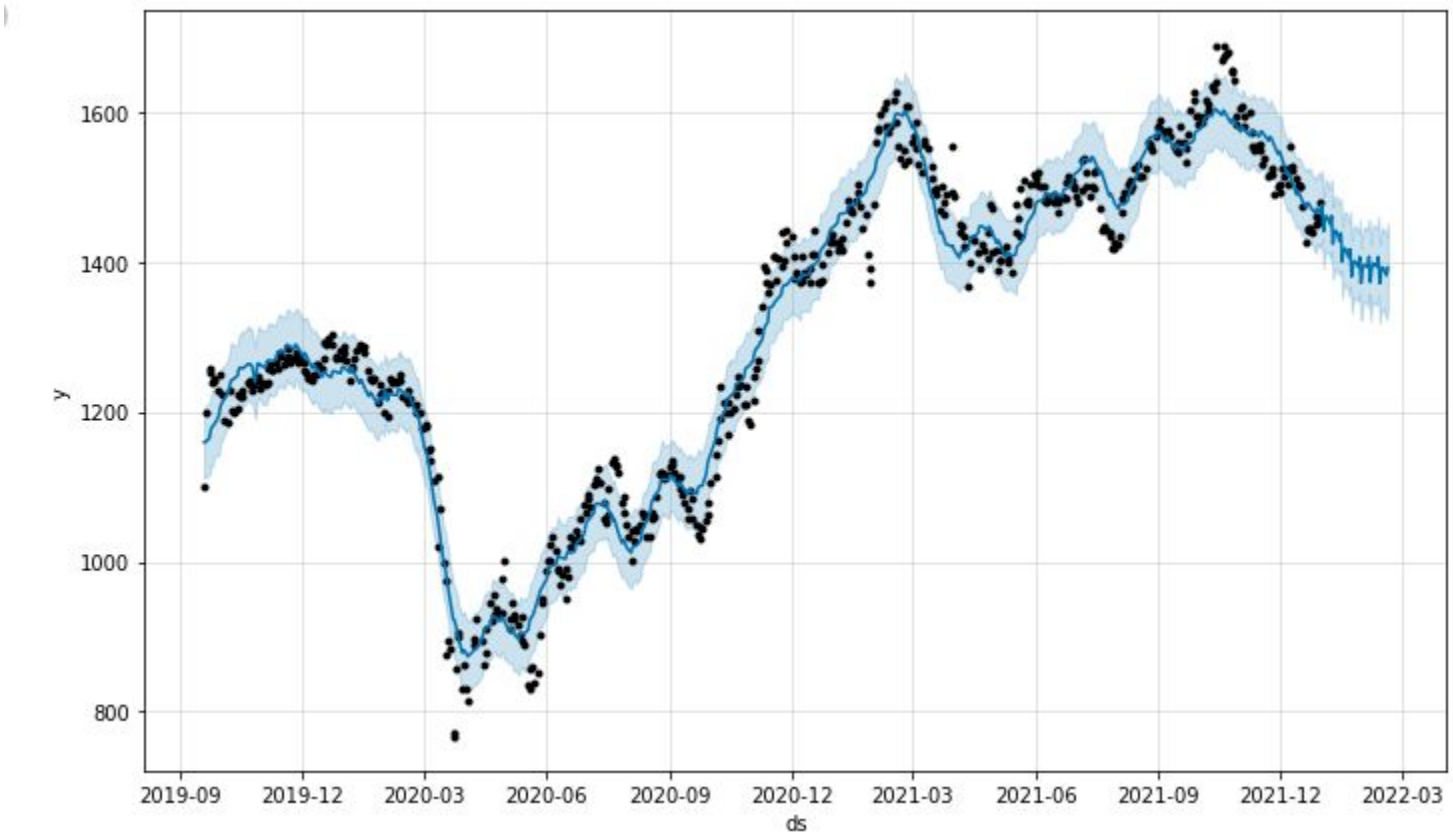
Model_Name	MSE	RMSE	MAE
ARIMA	5325.129870	72.973487	54.133885
SARIMA	4135.333795	64.306561	49.791887

PROPHET

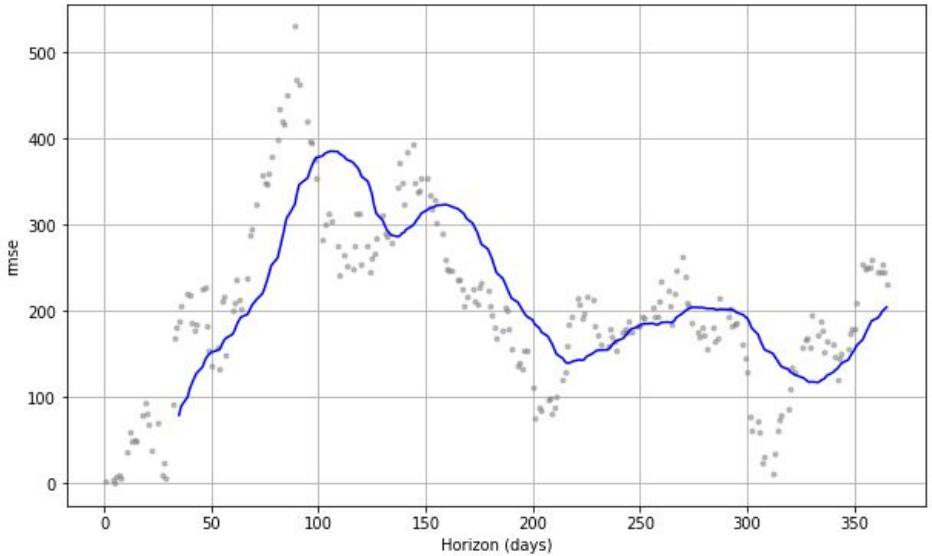
Prophet

	ds	yhat	yhat_lower	yhat_upper	y
243	2021-12-27	1695.934495	1450.367449	1942.051595	1450.80
244	2021-12-28	1705.417733	1453.918485	1947.968916	1460.80
245	2021-12-29	1708.874098	1458.622114	1952.116442	1453.85
246	2021-12-30	1707.696148	1455.759018	1961.029989	1461.50
247	2021-12-31	1709.647145	1449.435518	1960.311169	1479.40

Forecast:


1600
1400
1200
1000
800
y
2019-09
2019-12
2020-03
2020-06
2020-09
2020-12
2021-03
2021-06
2021-09
2021-12
2022-03
ds

Cross Validation metric



Performance Metrics

Model_Name	MSE	MAE	MAPE
Fb Prophet	50860.367126	202.615891	13.460252

DEPLOYMENT

Thank You!

Contact us

Prasanna Pprakash | LinkedIn

Dhanvanthri | LinkedIn

Boobesh | LinkedIn