In [1]:
```
import pyforest
```

In [2]:
```
df = pd.read_csv("HCLTECH.csv")
```

In [3]:
```
df
```

Out[3]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWA
0	2000-01-11	HCLTECH	EQ	580.00	1550.0	1725.00	1492.00	1560.00	1554.45	1582.7
1	2000-01-12	HCLTECH	EQ	1554.45	1560.0	1678.85	1560.00	1678.85	1678.85	1657.0
2	2000-01-13	HCLTECH	EQ	1678.85	1790.0	1813.20	1781.00	1813.20	1813.20	1804.6
3	2000-01-14	HCLTECH	EQ	1813.20	1958.3	1958.30	1835.00	1958.30	1958.30	1939.9
4	2000-01-17	HCLTECH	EQ	1958.30	2115.0	2115.00	1801.65	1801.65	1801.65	1990.5
...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.0	954.50	923.05	930.00	928.80	931.7
5296	2021-04-27	HCLTECH	EQ	928.80	931.0	938.55	923.40	930.30	928.85	928.0
5297	2021-04-28	HCLTECH	EQ	928.85	931.2	935.85	921.75	925.90	923.80	926.6
5298	2021-04-29	HCLTECH	EQ	923.80	929.7	929.70	907.10	910.30	909.55	914.3
5299	2021-04-30	HCLTECH	EQ	909.55	905.0	915.00	895.40	900.10	898.95	904.9

5300 rows × 15 columns

In [4]:
```
df = df[(df['Date'] > '2014-12-31')]
```

In [5]:
```
df
```

Out[5]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VW
3734	2015-01-01	HCLTECH	EQ	1596.90	1599.00	1611.65	1585.60	1605.50	1606.80	1604.
3735	2015-01-02	HCLTECH	EQ	1606.80	1602.05	1618.95	1600.05	1607.00	1605.25	1608.
3736	2015-01-05	HCLTECH	EQ	1605.25	1615.00	1615.00	1568.70	1581.00	1578.25	1586.
3737	2015-01-06	HCLTECH	EQ	1578.25	1574.00	1574.40	1522.65	1543.05	1536.10	1553.
3738	2015-01-07	HCLTECH	EQ	1536.10	1539.50	1548.00	1492.70	1500.00	1499.60	1517.
...	...	...	...	...	...	...	...	...	...	
5295	2021-04-26	HCLTECH	EQ	955.65	940.00	954.50	923.05	930.00	928.80	931.
5296	2021-04-27	HCLTECH	EQ	928.80	931.00	938.55	923.40	930.30	928.85	928.
5297	2021-04-28	HCLTECH	EQ	928.85	931.20	935.85	921.75	925.90	923.80	926.
5298	2021-04-29	HCLTECH	EQ	923.80	929.70	929.70	907.10	910.30	909.55	914.
5299	2021-04-30	HCLTECH	EQ	909.55	905.00	915.00	895.40	900.10	898.95	904.

1566 rows × 15 columns

In [6]:

```
df = df[["Date","VWAP"]]
# Rename the features: These names are required for the model fitting
df = df.rename(columns = {"Date":"ds","VWAP":"y"})
df.head()
```

Out[6]:

	ds	y
3734	2015-01-01	1604.76
3735	2015-01-02	1608.73
3736	2015-01-05	1586.98
3737	2015-01-06	1553.35
3738	2015-01-07	1517.47

In [7]:
```
df.reset_index()
```

Out[7]:

	index	ds	y
0	3734	2015-01-01	1604.76
1	3735	2015-01-02	1608.73
2	3736	2015-01-05	1586.98
3	3737	2015-01-06	1553.35
4	3738	2015-01-07	1517.47
...	...	...	...
1561	5295	2021-04-26	931.70
1562	5296	2021-04-27	928.06
1563	5297	2021-04-28	926.63
1564	5298	2021-04-29	914.34
1565	5299	2021-04-30	904.98

1566 rows × 3 columns

In [8]:
```
df
```

Out[8]:

	ds	y
3734	2015-01-01	1604.76
3735	2015-01-02	1608.73
3736	2015-01-05	1586.98
3737	2015-01-06	1553.35
3738	2015-01-07	1517.47
...	...	...
5295	2021-04-26	931.70
5296	2021-04-27	928.06
5297	2021-04-28	926.63
5298	2021-04-29	914.34
5299	2021-04-30	904.98

1566 rows × 2 columns

In [11]:
```
from fbprophet import Prophet
# The Prophet class (model)
fbp = Prophet(daily_seasonality = True)
 # Fit the model
fbp.fit(df)
# We need to specify the number of days in future
# We'll be predicting next 2 mont stock prices
fut = fbp.make_future_dataframe(periods=60)
forecast = fbp.predict(fut)
```

```
Importing plotly failed. Interactive plots will not work.
INFO:numexpr.utils:NumExpr defaulting to 4 threads.
```

In [15]:
```
forecast[(forecast['ds'] > '2021-04-30')]['yhat']
```

Out[15]:
```
1566    1024.348662
1567     988.928542
1568     905.198913
1569     903.151135
1570     903.894547
1571     900.026771
1572     899.077930
1573    1017.301157
1574     984.509589
1575     903.608717
1576     904.518234
1577     908.274262
1578     907.401601
1579     909.359840
1580    1030.335647
1581    1000.081396
1582     921.448963
1583     924.312796
1584     929.673099
```

```
1585     930.028773
1586     932.823928
1587    1054.240274
1588    1024.035574
1589     945.076976
1590     947.263366
1591     951.627602
1592     950.708305
1593     951.995053
1594    1071.719344
1595    1039.691526
1596     958.832585
1597     959.094957
1598     961.563284
1599     958.824814
1600     958.413347
1601    1076.599107
1602    1043.224785
1603     961.236652
1604     960.604917
1605     962.424282
1606     959.284367
1607     958.713911
1608    1076.971126
1609    1043.880405
1610     962.364282
1611     962.364777
1612     964.945038
1613     962.660939
1614     963.006476
1615    1082.205308
1616    1050.048633
1617     969.428211
1618     970.258682
1619     973.580035
1620     971.929872
1621     972.789295
1622    1092.374746
1623    1060.475972
1624     979.988957
1625     980.837943
Name: yhat, dtype: float64
```

In [16]:
```
forecast
```

Out[16]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_te
0	2015-01-01	1632.295891	1644.109801	1908.047571	1632.295891	1632.295891	146.674
1	2015-01-02	1627.178943	1639.347819	1905.411681	1627.178943	1627.178943	146.276
2	2015-01-05	1611.828098	1630.334126	1890.020560	1611.828098	1611.828098	150.307
3	2015-01-06	1606.711150	1632.194599	1890.588141	1606.711150	1606.711150	150.421
4	2015-01-07	1601.594201	1621.094736	1892.125501	1601.594201	1601.594201	153.137
...	...	...	...	...	...	...	
1621	2021-06-25	853.763418	847.103296	1127.527937	842.273241	868.146795	119.025
1622	2021-06-26	854.690727	964.020579	1227.304333	843.425131	869.449212	237.684
1623	2021-06-27	855.618036	934.401627	1193.291545	844.223566	870.884808	204.857
1624	2021-06-28	856.545345	852.666250	1124.158336	844.908213	872.103370	123.443
1625	2021-06-29	857.472654	835.954209	1118.711557	845.522162	874.447617	123.365

1626 rows × 22 columns

In [ ]: