Stock Price prediction using Facebook Prophet

```
## Switch to GPU mode for faster Computation (Runtime> Change runtime> GPU)
```

Importing all the necessary Libraries

```
#Necessary libraries = Pandas, fbprophet and plotly

#pandas= data Manipulation and analysis
#fbprophet = Forecasting
#plotly= data visualization
```

```
!pip install prophet
!pip install yfinance
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: prophet in /usr/local/lib/python3.8/dist-packages (1.1.2)
Requirement already satisfied: holidays>=0.14.2 in /usr/local/lib/python3.8/dist-packages (from prophet) (0.19)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.8/dist-packages (from prophet) (4.64.1)
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.8/dist-packages (from prophet) (2.8.2)
Requirement already satisfied: LunarCalendar>=0.0.9 in /usr/local/lib/python3.8/dist-packages (from prophet) (0.0.9)
Requirement already satisfied: matplotlib>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from prophet) (3.2.2)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.8/dist-packages (from prophet) (1.21.6)
Requirement already satisfied: cmdstanpy>=1.0.4 in /usr/local/lib/python3.8/dist-packages (from prophet) (1.1.0)
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.8/dist-packages (from prophet) (1.3.5)
Requirement already satisfied: convertdate>=2.1.2 in /usr/local/lib/python3.8/dist-packages (from prophet) (2.4.0)
Requirement already satisfied: pymeeus<=1,>=0.3.13 in /usr/local/lib/python3.8/dist-packages (from convertdate>=2.1.2->prophet) (0.5.12)
Requirement already satisfied: hijri-converter in /usr/local/lib/python3.8/dist-packages (from holidays>=0.14.2->prophet) (2.2.4)
Requirement already satisfied: korean-lunar-calendar in /usr/local/lib/python3.8/dist-packages (from holidays>=0.14.2->prophet) (0.3.1)
Requirement already satisfied: ephem>=3.7.5.3 in /usr/local/lib/python3.8/dist-packages (from LunarCalendar>=0.0.9->prophet) (4.1.4)
Requirement already satisfied: pytz in /usr/local/lib/python3.8/dist-packages (from LunarCalendar>=0.0.9->prophet) (2022.7.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.0.0
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.0.0->prophet) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.0.0->prophet) (0.11.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.8.0->prophet) (1.15.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: yfinance in /usr/local/lib/python3.8/dist-packages (0.2.9)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (4.9.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2.3.4)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.1)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2022.7.1)
Requirement already satisfied: requests>=2.26 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2.28.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (4.11.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.21.6)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.3.5)
Requirement already satisfied: cryptography>=3.3.2 in /usr/local/lib/python3.8/dist-packages (from yfinance) (39.0.0)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.8/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.8/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.3.2.pos
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.8/dist-packages (from cryptography>=3.3.2->yfinance) (1.15.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.8/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.8/dist-packages (from html5lib>=1.1->yfinance) (1.15.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (1.24.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (2.1.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.8/dist-packages (from cffi>=1.12->cryptography>=3.3.2->yfinance) (2.21
```

```
import pandas as pd
import plotly.express as px
from prophet import Prophet
```

```
 #Initializing Plotly
import plotly.io as pio
pio.renderers.default='colab'
```

Importing the Dataset & Exploring it

```
df = pd.read_csv('TSLA1.csv')
```

```
#read_csv function from pandas
```

```
df.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	1.592667	281494500
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	1.588667	257806500
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	1.464000	123282000
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	1.280000	77097000
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	1.074000	103003500

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3173 entries, 0 to 3172
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       3173 non-null   object 
 1   Open       3173 non-null   float64
 2   High       3173 non-null   float64
 3   Low        3173 non-null   float64
 4   Close      3173 non-null   float64
 5   Adj Close  3173 non-null   float64
 6   Volume     3173 non-null   int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 173.6+ KB
```

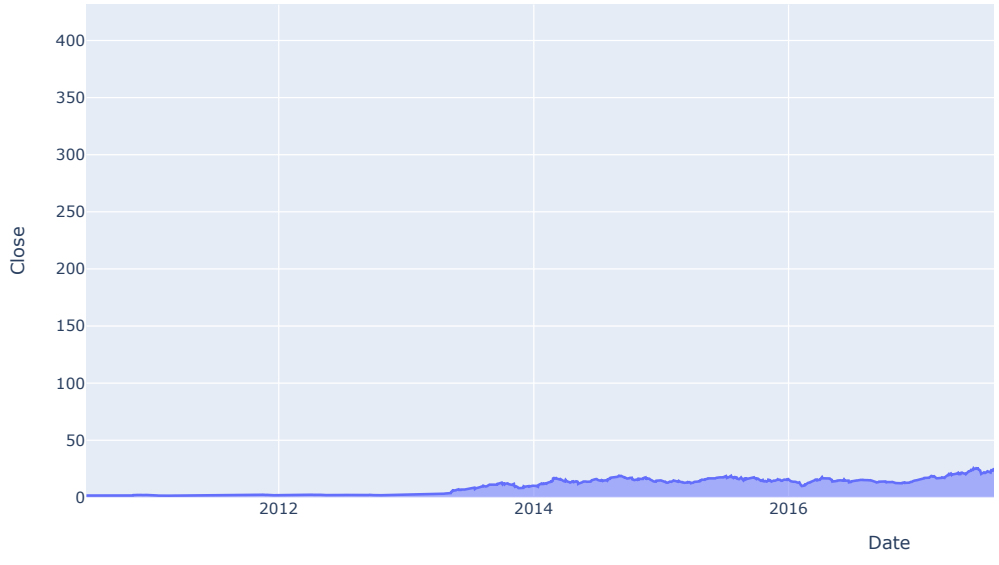
```
df.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	3173.000000	3173.000000	3173.000000	3173.000000	3173.000000	3.173000e+03
mean	59.439744	60.788224	57.965386	59.403464	59.403464	9.435458e+07
std	95.575692	97.789307	93.094113	95.460596	95.460596	8.193729e+07
min	1.076000	1.108667	0.998667	1.053333	1.053333	1.777500e+06
25%	9.123333	9.446667	8.940667	9.186667	9.186667	4.260090e+07
50%	16.336666	16.544666	16.083332	16.316000	16.316000	7.619220e+07
75%	25.199333	25.666668	24.846666	25.304667	25.304667	1.187910e+08
max	411.470001	414.496674	405.666656	409.970001	409.970001	9.140820e+08

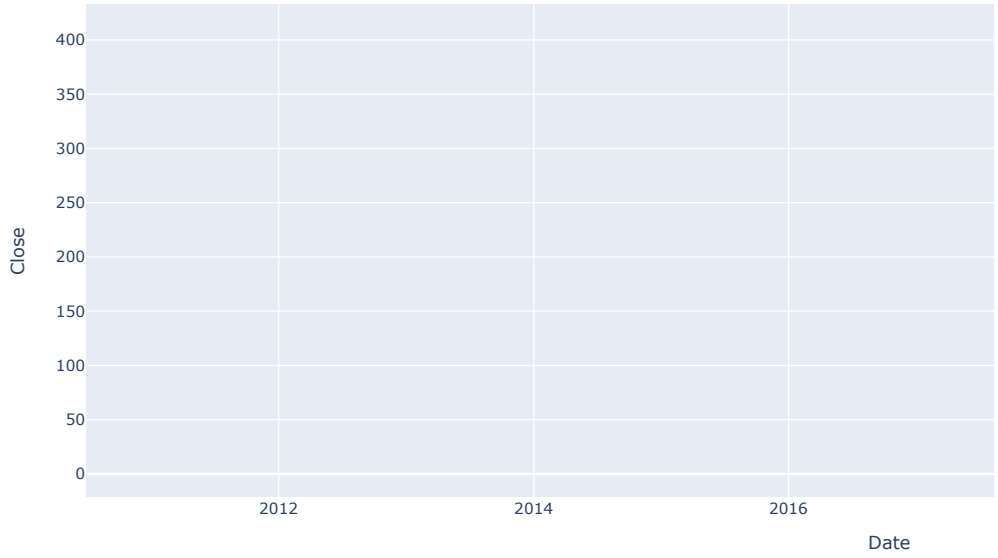
Data Visualization using plotly express- Visualizing the historical performance of Tesla

```
#Line graph, Area graph , box plot (Analyzing price and volume)
```

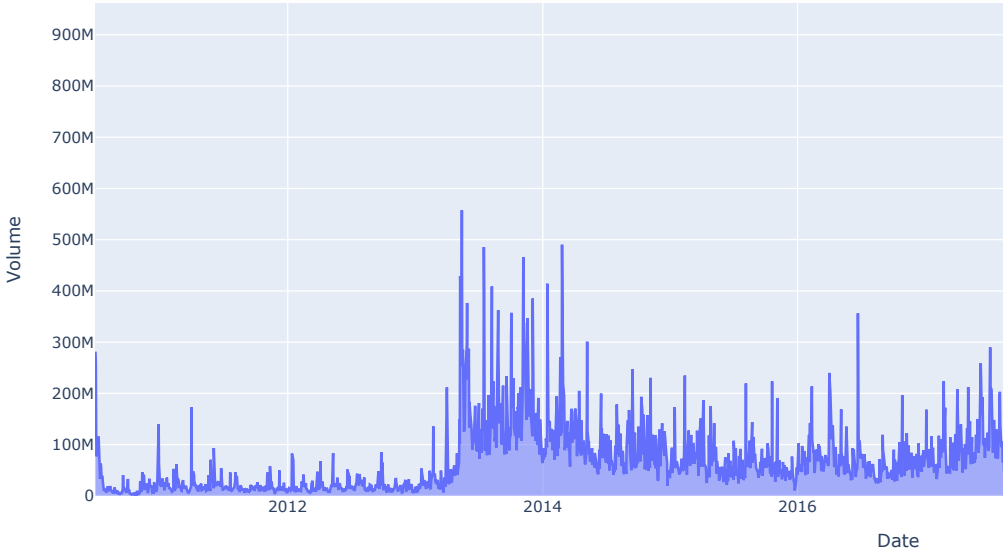
```
px.area(df, x="Date", y="Close")
```



```
px.line(df, x="Date", y="Close")
```

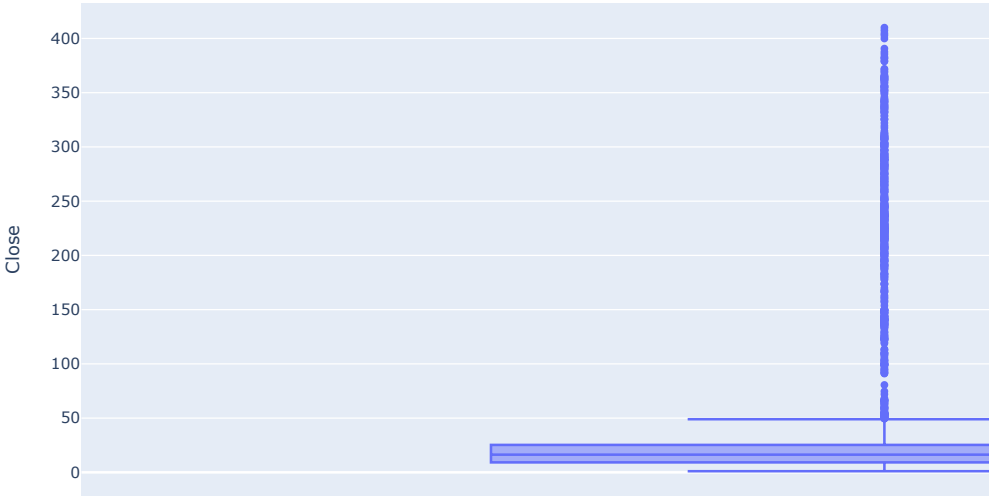



```
px.area(df, x="Date", y="Volume")
```

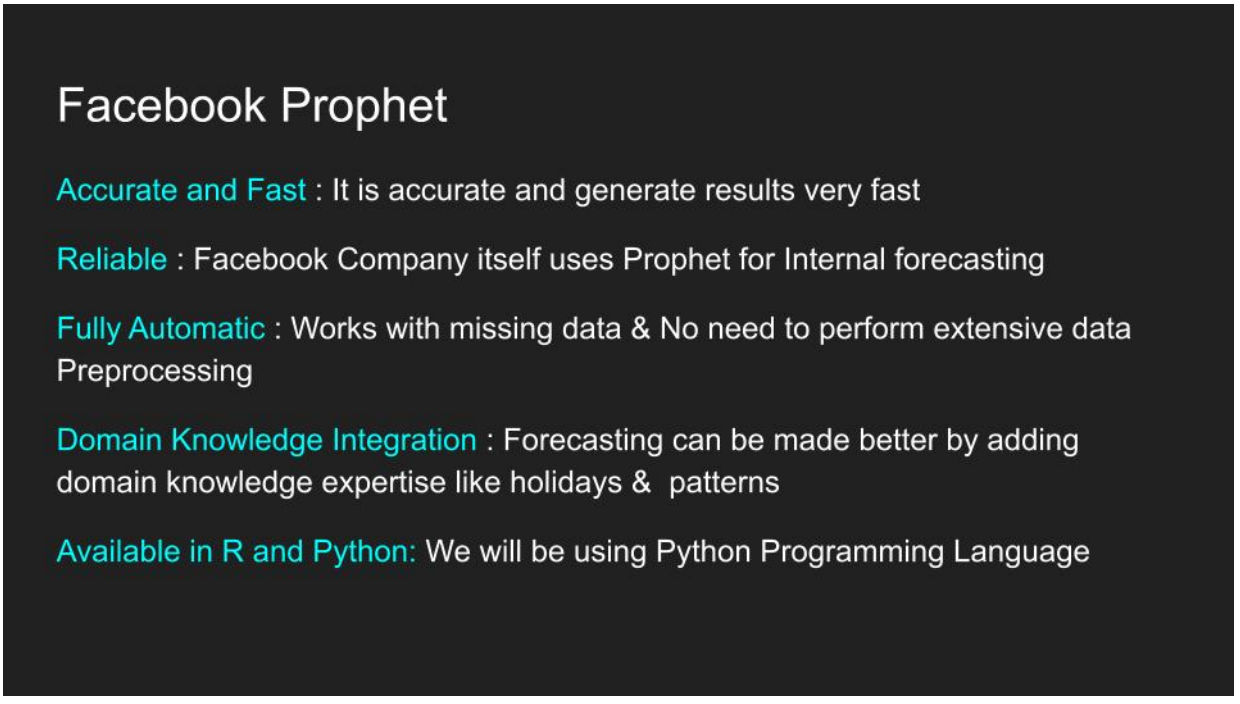



```
px.bar(df, y="Volume")
```


```
px.box(df, y="Close")
```



Understanding Facebook Prophet



Data Preperation

```
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	1.592667	281494500
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	1.588667	257806500
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	1.464000	123282000
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	1.280000	77097000
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	1.074000	103003500
...	...	...	...	...	...	...	...
3168	2023-01-30	178.050003	179.770004	166.500000	166.660004	166.660004	230878800
3169	2023-01-31	164.570007	174.300003	162.779999	173.220001	173.220001	196813500
3170	2023-02-01	173.889999	183.809998	169.929993	181.410004	181.410004	213806300

```
col=['Date','Close']
ndf=pd.DataFrame(df,columns=col)
```

```
ndf
```

	Date	Close
0	2010-06-29	1.592667
1	2010-06-30	1.588667
2	2010-07-01	1.464000
3	2010-07-02	1.280000
4	2010-07-06	1.074000
...	...	...
3168	2023-01-30	166.660004
3169	2023-01-31	173.220001
3170	2023-02-01	181.410004
3171	2023-02-02	188.270004
3172	2023-02-03	189.979996

3173 rows × 2 columns

```
prophet_df=ndf.rename(columns={'Date':'ds','Close':'y'})
```

```
prophet_df
```

	ds	y
0	2010-06-29	1.592667
1	2010-06-30	1.588667
2	2010-07-01	1.464000
3	2010-07-02	1.280000
4	2010-07-06	1.074000
...	...	...
3168	2023-01-30	166.660004
3169	2023-01-31	173.220001
3170	2023-02-01	181.410004
3171	2023-02-02	188.270004
3172	2023-02-03	189.979996

3173 rows × 2 columns

Creating Facebook Prophet Model

```
m=Prophet()
m.fit(prophet_df)
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpswtiuulw/yppsmq0u.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpswtiuulw/xz69v2vy.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.8/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=61677', 'da
```

```
14:08:09 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
14:08:10 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7f104f746490>
```

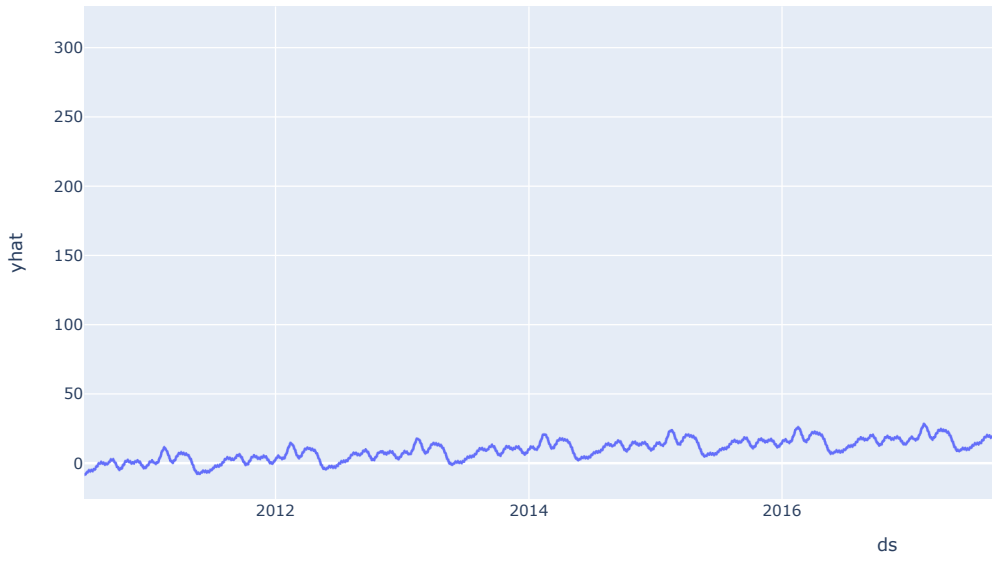
Forecasting

```
future=m.make_future_dataframe(periods=30)
forecast=m.predict(future)
```

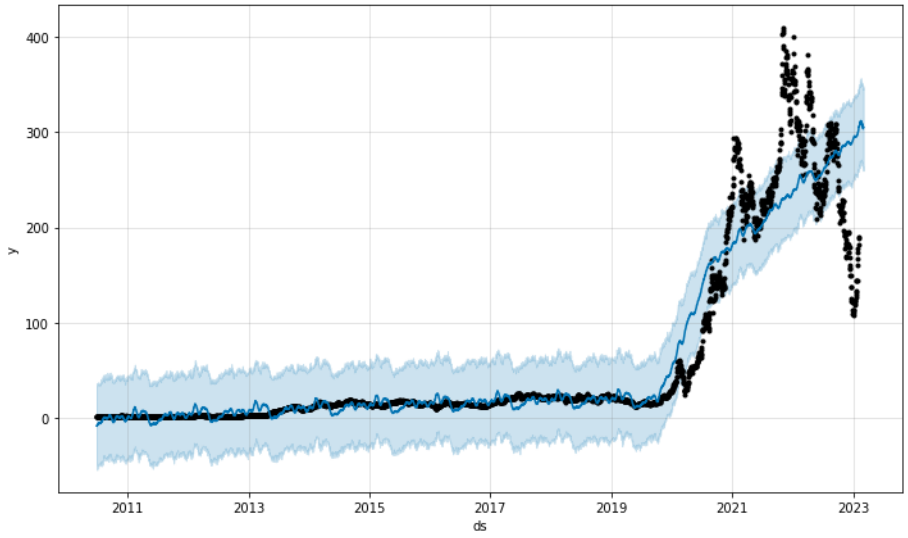
```
forecast
```

ıdditive_terms_upper	weekly	weekly_lower	weekly_upper	yearly	yearly_lower	yearly_upper	multipl
-7.140419	-0.584796	-0.584796	-0.584796	-6.555623	-6.555623	-6.555623	
-6.770310	-0.457741	-0.457741	-0.457741	-6.312570	-6.312570	-6.312570	
-6.791260	-0.728528	-0.728528	-0.728528	-6.062732	-6.062732	-6.062732	
-6.917047	-1.106144	-1.106144	-1.106144	-5.810902	-5.810902	-5.810902	
-5.460753	-0.584796	-0.584796	-0.584796	-4.875956	-4.875956	-4.875956	
...	...	...	...	...	...	...	
2.920551	-0.457741	-0.457741	-0.457741	3.378292	3.378292	3.378292	
2.001281	-0.728528	-0.728528	-0.728528	2.729809	2.729809	2.729809	
1.026942	-1.106144	-1.106144	-1.106144	2.133086	2.133086	2.133086	
2.965718	1.367236	1.367236	1.367236	1.598482	1.598482	1.598482	
2.501977	1.367235	1.367235	1.367235	1.134742	1.134742	1.134742	

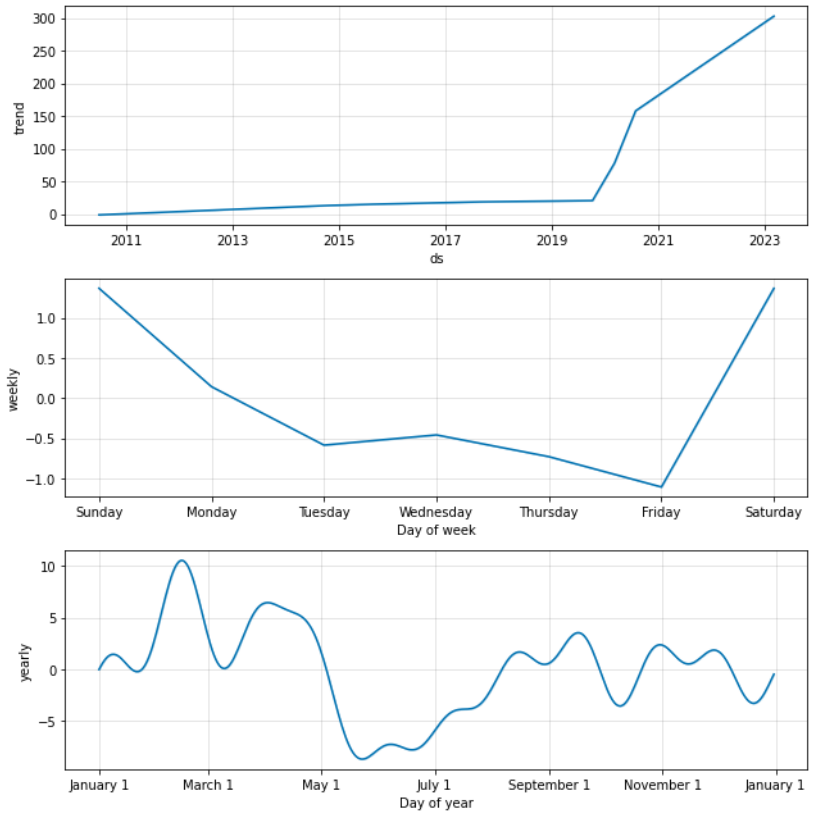
```
px.line(forecast, x='ds', y='yhat')
```




```
figure=m.plot(forecast, xlabel='ds', ylabel='y')
```




```
figure2=m.plot_components(forecast)
```



Downloading the Forecast data

```
from google.colab import files
forecast.to_csv('forecast.csv')
#files.download('forecast.csv')
```

✓ 0s completed at 19:38

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.