Amazon-AMZNStockPricePredictionUsingFacebookProphet

September 5, 2025

1 Amazon (AMZN) Stock Price prediction using Facebook Prophet

[1]:
```
## Switch to GPU mode for faster Computation (Runtime> Change runtime> GPU)
```

1.1 Importing all the necessary Libraries

[2]:
```
#Necessary libraries = Pandas, fbprophet and plotly

#pandas= data Manipulation and analysis
#fbprophet = Forecasting
#plotly= data visualization
#!pip install prophet
```

[3]:
```
import pandas as pd
import plotly.express as px
#from fbprophet import Prophet
from prophet import Prophet
#
#import ipywidgets as widgets
%matplotlib inline
```

[4]:
```
 #Initializing Plotly
import plotly.io as pio
pio.renderers.default='colab'
```

1.2 Importing the Dataset & Exploring it

[5]:
```
df = pd.read_csv('AMZN.csv')
```

[6]:
```
#read_csv function from pandas
```

[7]:
```
df
```

[7]:
```
           Date       Open       High        Low      Close  Adj Close  \
0    2022-12-12  89.209999  90.580002  87.870003  90.550003  90.550003
1    2022-12-13  95.230003  96.250000  90.519997  92.489998  92.489998
```

```
2    2022-12-14   92.500000   93.459999   89.870003   91.580002   91.580002
3    2022-12-15   89.889999   89.970001   87.470001   88.449997   88.449997
4    2022-12-16   88.269997   89.349998   86.730003   87.860001   87.860001
..          …           …           …           …           …           …
247  2023-12-06  147.580002  147.850006  144.279999  144.520004  144.520004
248  2023-12-07  146.149994  147.919998  145.339996  146.880005  146.880005
249  2023-12-08  145.479996  147.839996  145.399994  147.419998  147.419998
250  2023-12-11  145.660004  146.190002  143.639999  145.889999  145.889999
251  2023-12-12  145.520004  147.500000  145.300003  147.479996  147.479996

        Volume
0     61999800
1    100212000
2     70298000
3     84802900
4    146144100
..         …
247   39679000
248   52352800
249   41858000
250   50907300
251   44886600

[252 rows x 7 columns]
```

```
[8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252 entries, 0 to 251
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       252 non-null    object
 1   Open       252 non-null    float64
 2   High       252 non-null    float64
 3   Low        252 non-null    float64
 4   Close      252 non-null    float64
 5   Adj Close  252 non-null    float64
 6   Volume     252 non-null    int64
dtypes: float64(5), int64(1), object(1)
memory usage: 13.9+ KB
```

```
[9]: df.describe()
```

```
[9]:          Open        High         Low       Close   Adj Close  \
count  252.000000  252.000000  252.000000  252.000000  252.000000
mean   117.831072  119.368730  116.353016  117.966469  117.966469
std     19.125679   19.142427   19.115752   19.131681   19.131681
```

```
min      82.800003   83.480003   81.430000   81.820000   81.820000
25%      99.725001  101.044998   98.094999  100.010000  100.010000
50%     124.774998  126.730003  123.689999  125.000000  125.000000
75%     133.597504  134.699997  131.982506  133.334996  133.334996
max     147.850006  149.259995  146.880005  147.729996  147.729996

             Volume
count  2.520000e+02
mean   6.029315e+07
std    2.135646e+07
min    2.237840e+07
25%    4.643740e+07
50%    5.564410e+07
75%    6.792652e+07
max    1.581542e+08
```

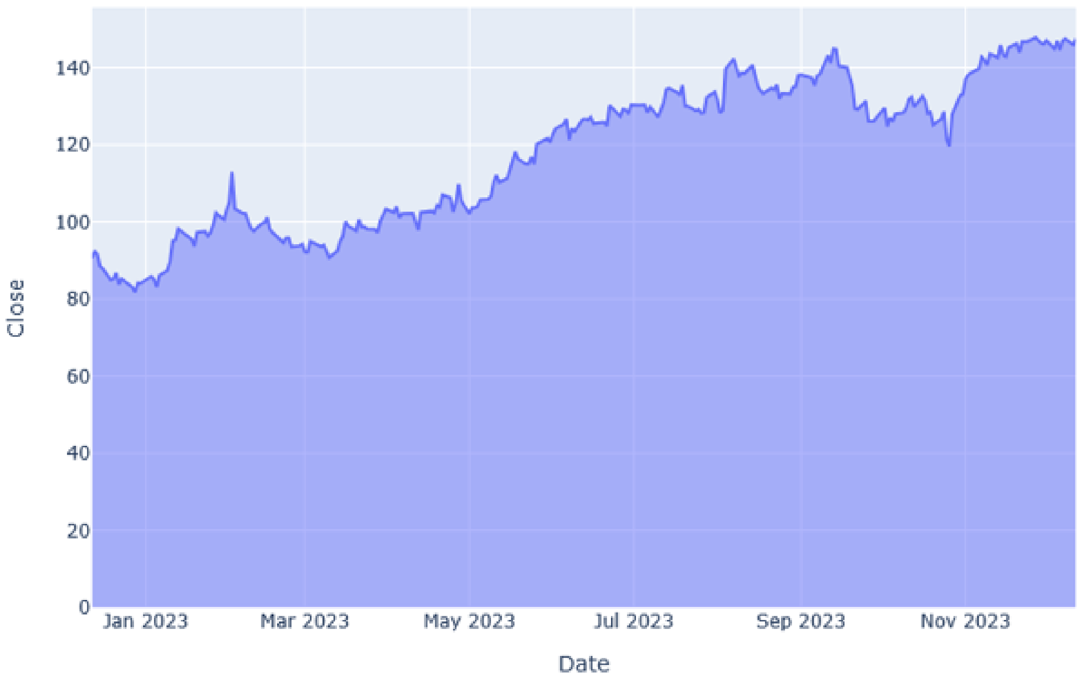
2 Data Visualization using plotly express- Visualizing the historical performance of Amazon

[10]:

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

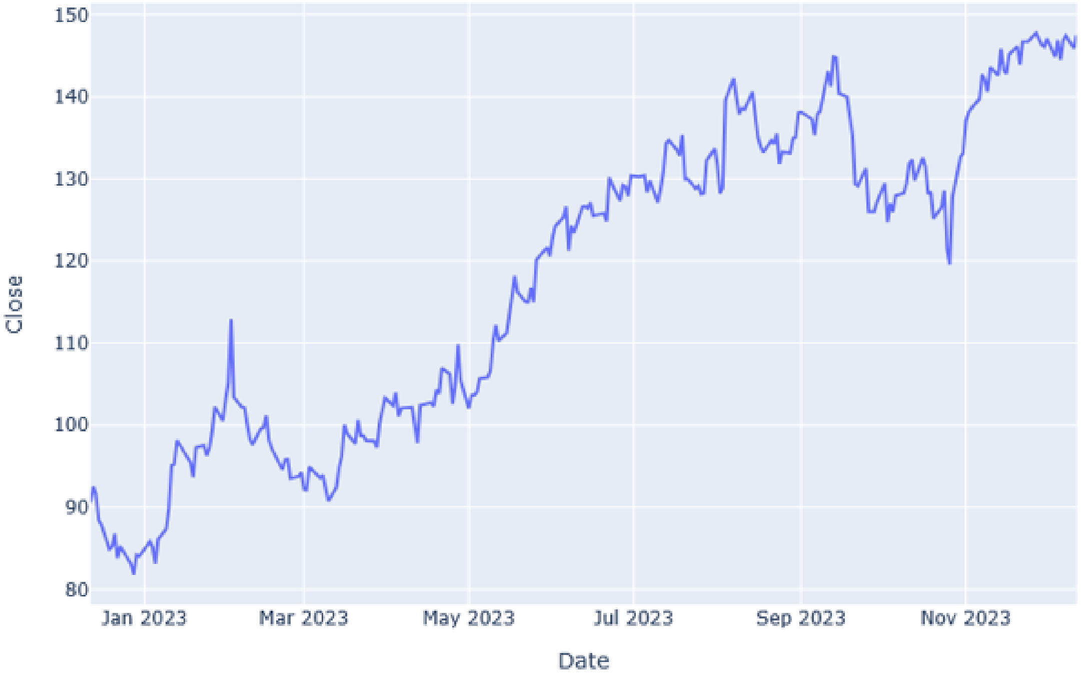
[33]:

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

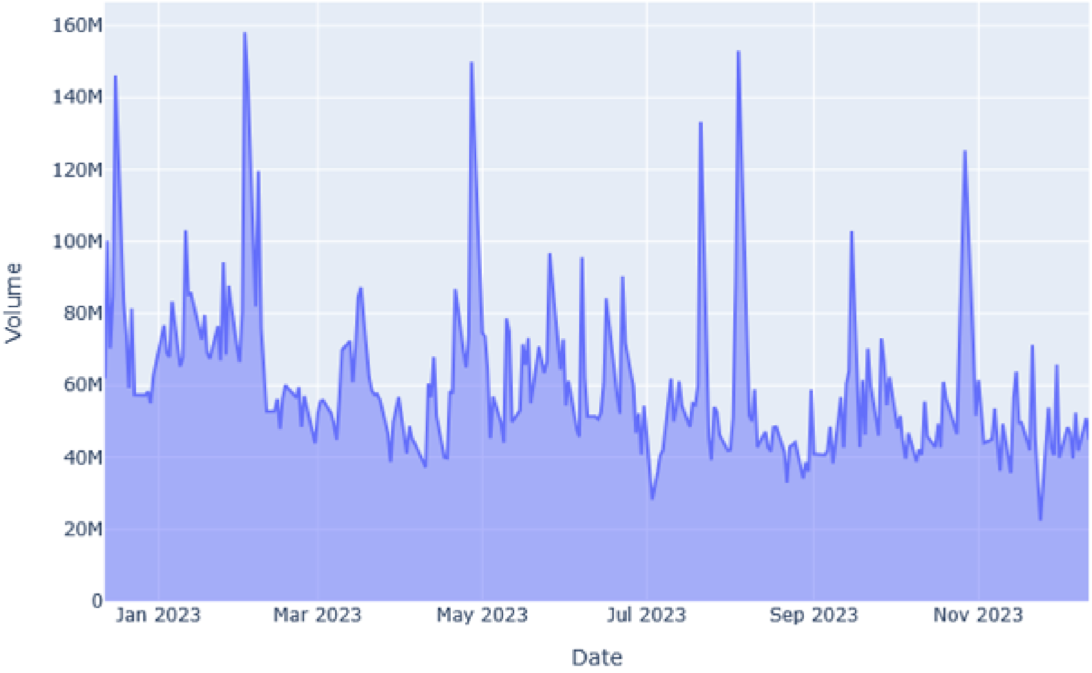


[34]:

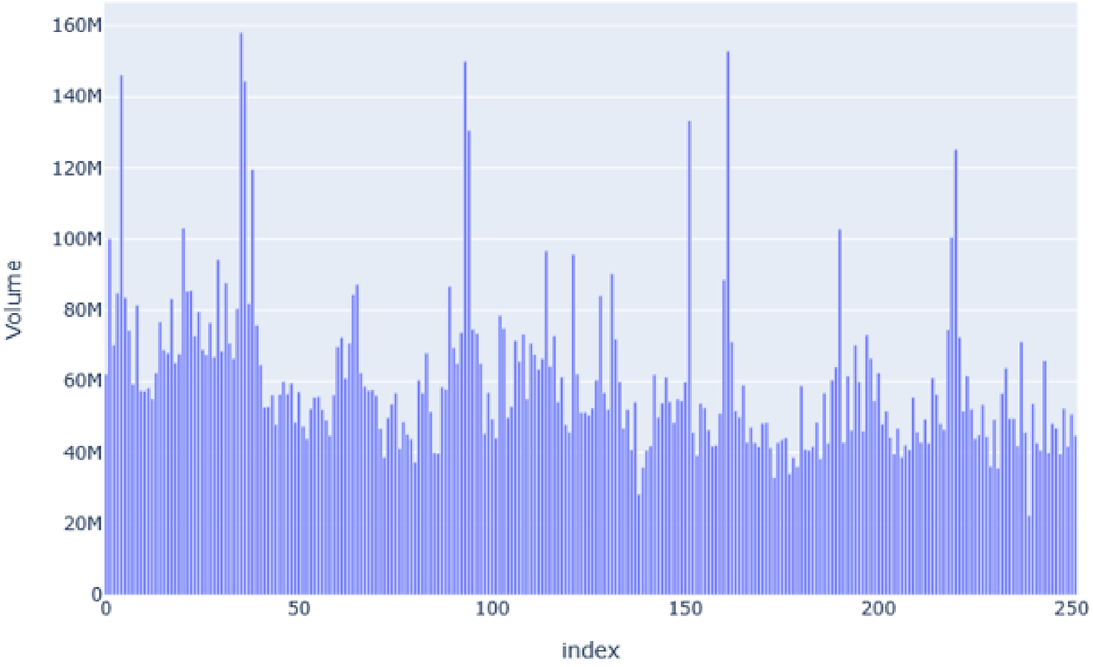
```
px.line(df, x='Date',y='Close')
```



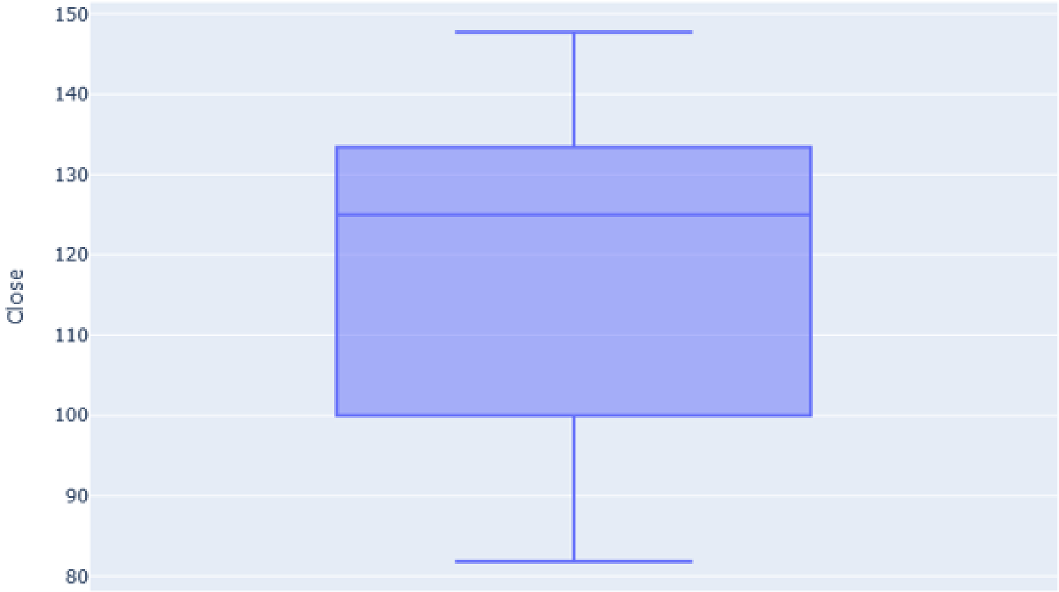
[35]:
```
px.area(df, x='Date',y='Volume')
```



[36]:
```
px.bar(df, y='Volume')
```



[37]:
```
px.box(df, y='Close')
```



3 Understanding Facebook Prophet

3.1 Data Preparation

[16]:
```
df
```

[16]:
```
           Date        Open        High         Low       Close   Adj Close  \
0    2022-12-12   89.209999   90.580002   87.870003   90.550003   90.550003
1    2022-12-13   95.230003   96.250000   90.519997   92.489998   92.489998
2    2022-12-14   92.500000   93.459999   89.870003   91.580002   91.580002
3    2022-12-15   89.889999   89.970001   87.470001   88.449997   88.449997
4    2022-12-16   88.269997   89.349998   86.730003   87.860001   87.860001
..          …           …           …           …           …           …
247  2023-12-06  147.580002  147.850006  144.279999  144.520004  144.520004
248  2023-12-07  146.149994  147.919998  145.339996  146.880005  146.880005
249  2023-12-08  145.479996  147.839996  145.399994  147.419998  147.419998
250  2023-12-11  145.660004  146.190002  143.639999  145.889999  145.889999
251  2023-12-12  145.520004  147.500000  145.300003  147.479996  147.479996

        Volume
0     61999800
1    100212000
2     70298000
3     84802900
4    146144100
..         …
247   39679000
248   52352800
249   41858000
250   50907300
251   44886600

[252 rows x 7 columns]
```

[17]:
```
  columns = ['Date','Close']
```

[18]:
```
ndf = pd.DataFrame(df, columns = columns)
```

[19]:
```
ndf
```

[19]:
```
           Date       Close
0    2022-12-12   90.550003
1    2022-12-13   92.489998
2    2022-12-14   91.580002
3    2022-12-15   88.449997
4    2022-12-16   87.860001
..          …           …
247  2023-12-06  144.520004
248  2023-12-07  146.880005
249  2023-12-08  147.419998
```

```
250  2023-12-11  145.889999
251  2023-12-12  147.479996

[252 rows x 2 columns]
```

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

```
[21]: prophet_df
```

```
[21]:             ds           y
      0    2022-12-12   90.550003
      1    2022-12-13   92.489998
      2    2022-12-14   91.580002
      3    2022-12-15   88.449997
      4    2022-12-16   87.860001
      ..          …          …
      247  2023-12-06  144.520004
      248  2023-12-07  146.880005
      249  2023-12-08  147.419998
      250  2023-12-11  145.889999
      251  2023-12-12  147.479996

      [252 rows x 2 columns]
```

3.2 Creating Facebook Prophet Model

```
[22]: m = Prophet()
      m.fit(prophet_df)
```

```
12:14:41 - cmdstanpy - INFO - Chain [1] start processing
12:14:42 - cmdstanpy - INFO - Chain [1] done processing
```

```
[22]: <prophet.forecaster.Prophet at 0x1f5ec93f0e0>
```

##Forecasting

```
[23]: future = m.make_future_dataframe(periods=30) # 30: 30 days
      future.tail()
```

```
[23]:            ds
      277 2024-01-07
      278 2024-01-08
      279 2024-01-09
      280 2024-01-10
      281 2024-01-11
```

```
[24]: forecast = m.predict(future)
      forecast
```

```
[24]:             ds       trend  yhat_lower  yhat_upper  trend_lower  trend_upper  \
      0   2022-12-12   87.742902   81.395116   93.433655    87.742902    87.742902
      1   2022-12-13   87.895196   80.979463   93.252805    87.895196    87.895196
      2   2022-12-14   88.047490   81.486475   93.189667    88.047490    88.047490
      3   2022-12-15   88.199784   81.260108   93.233233    88.199784    88.199784
      4   2022-12-16   88.352079   82.522873   93.919074    88.352079    88.352079
      ..         …         …         …         …          …          …
      277 2024-01-07  150.871199  146.631710  158.478209   149.787973   152.053351
      278 2024-01-08  151.055727  144.321301  156.609709   149.930003   152.308819
      279 2024-01-09  151.240254  144.632258  156.683386   150.050507   152.573912
      280 2024-01-10  151.424782  144.932691  156.645783   150.167805   152.840550
      281 2024-01-11  151.609309  144.511883  157.031792   150.277458   153.091756

           additive_terms  additive_terms_lower  additive_terms_upper    weekly  \
      0         -0.395304             -0.395304             -0.395304 -0.395304
      1         -0.523181             -0.523181             -0.523181 -0.523181
      2         -0.751175             -0.751175             -0.751175 -0.751175
      3         -0.691648             -0.691648             -0.691648 -0.691648
      4         -0.196727             -0.196727             -0.196727 -0.196727
      ..              …                   …                   …       …
      277        1.279018              1.279018              1.279018  1.279018
      278       -0.395304             -0.395304             -0.395304 -0.395304
      279       -0.523181             -0.523181             -0.523181 -0.523181
      280       -0.751175             -0.751175             -0.751175 -0.751175
      281       -0.691648             -0.691648             -0.691648 -0.691648

           weekly_lower  weekly_upper  multiplicative_terms  \
      0       -0.395304     -0.395304                   0.0
      1       -0.523181     -0.523181                   0.0
      2       -0.751175     -0.751175                   0.0
      3       -0.691648     -0.691648                   0.0
      4       -0.196727     -0.196727                   0.0
      ..            …           …                 …
      277      1.279018      1.279018                   0.0
      278     -0.395304     -0.395304                   0.0
      279     -0.523181     -0.523181                   0.0
      280     -0.751175     -0.751175                   0.0
      281     -0.691648     -0.691648                   0.0

           multiplicative_terms_lower  multiplicative_terms_upper        yhat
      0                           0.0                         0.0   87.347597
      1                           0.0                         0.0   87.372015
      2                           0.0                         0.0   87.296315
      3                           0.0                         0.0   87.508136
      4                           0.0                         0.0   88.155351
      ..                          …                         …         …
      277                         0.0                         0.0  152.150217
```

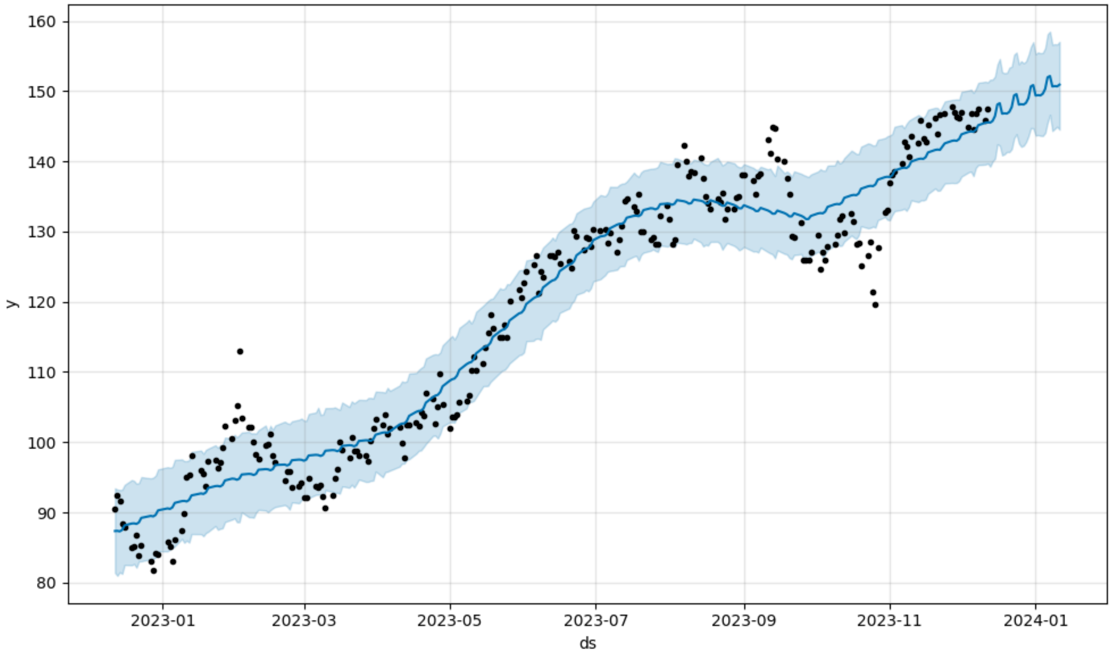
```
278                           0.0                           0.0  150.660423
279                           0.0                           0.0  150.717073
280                           0.0                           0.0  150.673607
281                           0.0                           0.0  150.917661

[282 rows x 16 columns]
```

```
[25]: forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

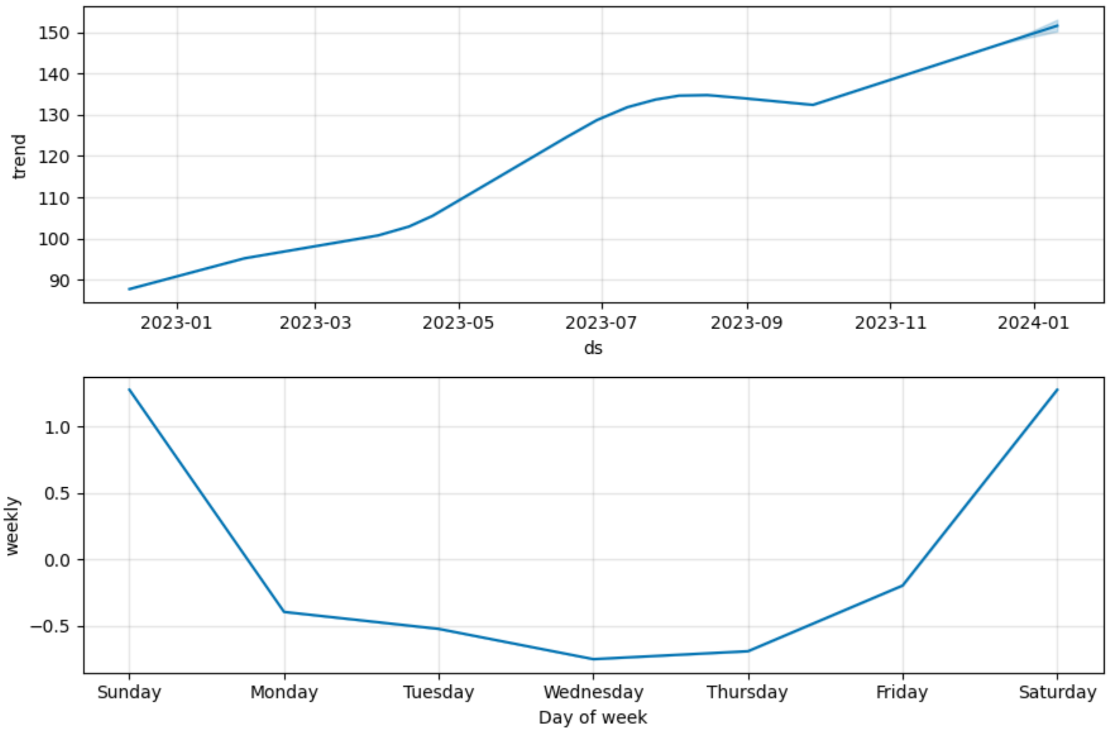
```
[25]:             ds        yhat  yhat_lower  yhat_upper
      277 2024-01-07  152.150217  146.631710  158.478209
      278 2024-01-08  150.660423  144.321301  156.609709
      279 2024-01-09  150.717073  144.632258  156.683386
      280 2024-01-10  150.673607  144.932691  156.645783
      281 2024-01-11  150.917661  144.511883  157.031792
```

```
[26]: px.line(forecast, x='ds', y='yhat')
```

```
[27]: figure = m.plot(forecast, xlabel='ds',ylabel = 'y')
```



```
[28]: figure2 = m.plot_components(forecast)
```



[29]:

```
from prophet.plot import plot_plotly, plot_components_plotly

plot_plotly(m, forecast)
```

[30]:

```
plot_components_plotly(m, forecast)
```

##Downloading the Forecast data

[31]:

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

[32]:

```
#forecast.to_csv('forecast.csv', index=False)
forecast.to_csv('forecast.csv', index=True)
```