```
import pandas as pd
import yfinance as yf
from datetime import datetime
from datetime import timedelta
import plotly.graph_objects as go
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly
import warnings
warnings.filterwarnings('ignore')
pd.options.display.float_format = '${:,.2f}'.format
today = datetime.today().strftime('%Y-%m-%d')
start_date = '2016-01-01'
eth_df = yf.download('ETH-USD',start_date, today)
eth_df.tail()
```

```
[*********************100%%**********************]  1 of 1 completed
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-09-12	$1,551.50	$1,619.11	$1,549.49	$1,592.43	$1,592.43	6813819740
2023-09-13	$1,592.89	$1,615.05	$1,582.22	$1,607.99	$1,607.99	4979469106
2023-09-14	$1,608.03	$1,640.52	$1,607.74	$1,626.97	$1,626.97	5538958553
2023-09-15	$1,626.87	$1,652.11	$1,613.25	$1,641.64	$1,641.64	4348584771
2023-09-16	$1,641.45	$1,649.99	$1,632.58	$1,635.22	$1,635.22	2819575929

```
eth_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2138 entries, 2017-11-09 to 2023-09-16
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Open       2138 non-null   float64
 1   High       2138 non-null   float64
 2   Low        2138 non-null   float64
 3   Close      2138 non-null   float64
 4   Adj Close  2138 non-null   float64
 5   Volume     2138 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 116.9 KB
```

```
eth_df.isnull().sum()
```

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

```
eth_df.columns
```

```
Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

```
eth_df.reset_index(inplace=True)
eth_df.columns
```

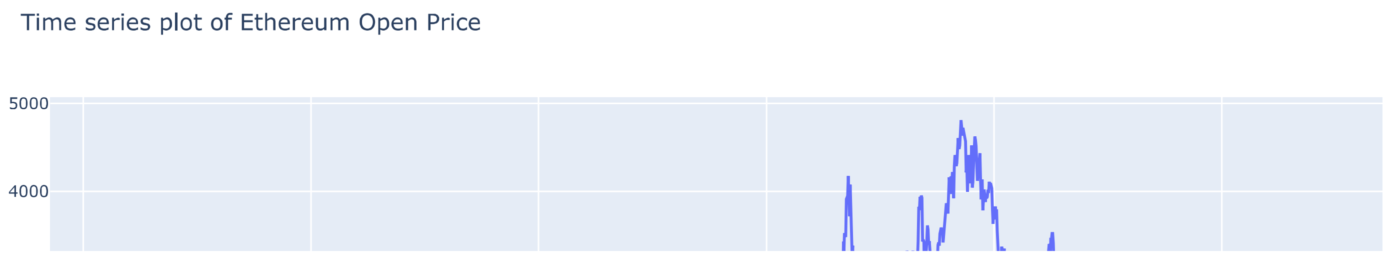
```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

```
df = eth_df[["Date", "Open"]]
new_names = {
    "Date": "ds",
    "Open": "y",
}
df.rename(columns=new_names, inplace=True)
```

```
df.tail()
```

	ds	y
2133	2023-09-12	$1,551.50
2134	2023-09-13	$1,592.89
2135	2023-09-14	$1,608.03
2136	2023-09-15	$1,626.87
2137	2023-09-16	$1,641.45

```
x = df["ds"]
y = df["y"]
fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=y))
# Set title
fig.update_layout(
    title_text="Time series plot of Ethereum Open Price",
)
```



```
m = Prophet(
    seasonality_mode="multiplicative"
)
m.fit(df)
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmp432_5bvl/_v5xoity.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp432_5bvl/srwbq7x0.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.10/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=33820', 'data', 'file=/tmp/tmp432_5bvl/_v5xoity.json',
10:12:37 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
10:12:38 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7de732a7a3b0>
```

```
future = m.make_future_dataframe(periods = 365)
future.tail()
```

	ds
2498	2024-09-11
2499	2024-09-12
2500	2024-09-13
2501	2024-09-14
2502	2024-09-15

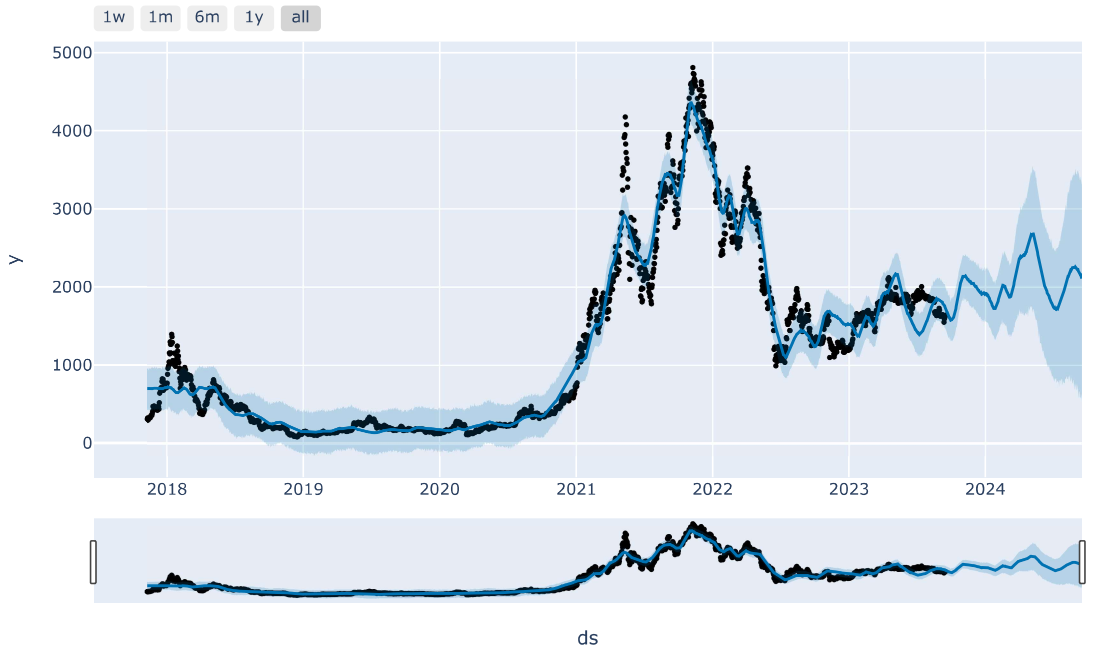
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
2498	2024-09-11	$2,161.05	$580.20	$3,372.64
2499	2024-09-12	$2,158.93	$579.73	$3,416.80
2500	2024-09-13	$2,137.39	$634.86	$3,364.12
2501	2024-09-14	$2,112.23	$603.99	$3,328.73
2502	2024-09-15	$2,105.61	$555.93	$3,310.50

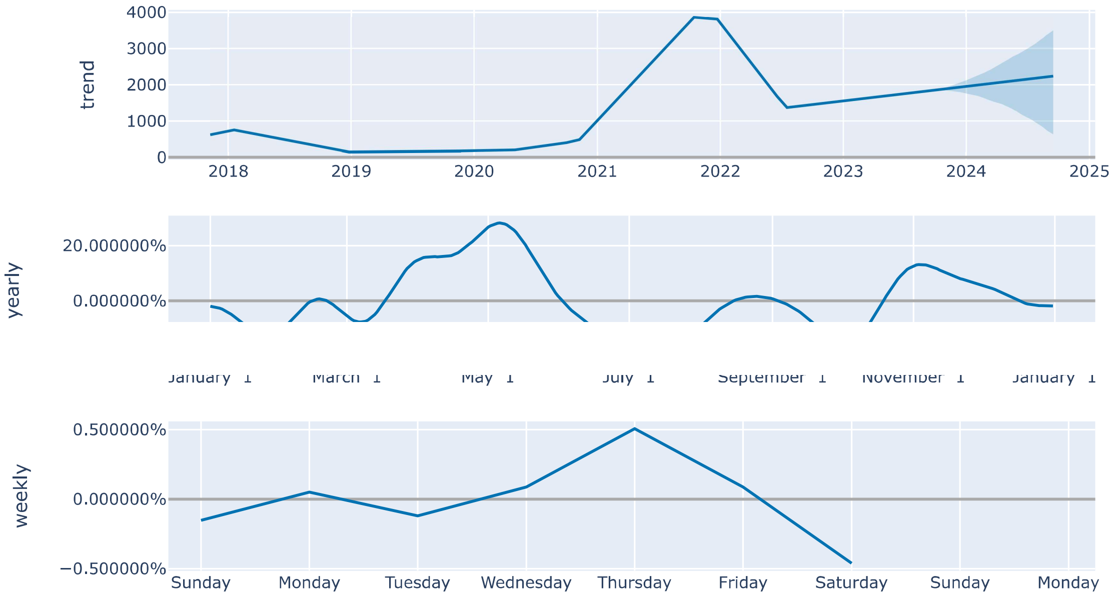
```
next_day = (datetime.today() + timedelta(days=1)).strftime('%Y-%m-%d')
forecast[forecast['ds'] == next_day]['yhat'].item()
```

```
1706.0104826583954
```

```
plot_plotly(m, forecast)
```



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


trend
4000
3000
2000
1000
0
2018
2019
2020
2021
2022
2023
2024
2025
yearly
20.000000%
0.000000%
January 1
March 1
May 1
July 1
September 1
November 1
January 1
weekly
0.500000%
0.000000%
−0.500000%
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Monday