Reference : https://facebook.github.io/prophet/docs/quick_start.html#python-api

```
from google.colab import drive
drive.mount('/gdrive')
```

```
    Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdr
```

```
%cd /gdrive/MyDrive/CMPE\ 272/Project\ data/hvac_ahu_sensors
```

```
    /gdrive/.shortcut-targets-by-id/1rkXHELIqJnWQplHIinDEuEUOHZF4x-yI/Project data/hvac_ahu_
```

```
import pandas as pd
import numpy as np
filename = 'ahu1_evac.csv'
data_raw = pd.read_csv(filename)
```

```
data_raw.head()
```

	Timestamp	demo/CTA1_Temperatura_Evacuare (°C)
0	06-Jan-17 5:15:00 PM EET	27.2
1	06-Jan-17 5:20:00 PM EET	27.2
2	06-Jan-17 5:25:00 PM EET	27.2
3	06-Jan-17 5:30:00 PM EET	27.2
4	06-Jan-17 5:35:00 PM EET	27.2

```
data_raw.Timestamp = data_raw.Timestamp.apply(lambda x: x[:-4])
```

```
data_raw.Timestamp = pd.to_datetime(data_raw.Timestamp)
```

```
data_raw = data_raw.rename(columns={"Timestamp": "ds", "demo/CTA1_Temperatura_Evacuare (°C)":
```

```
!pip install pystan==2.19.1.1 prophet
```

```
    Requirement already satisfied: pystan==2.19.1.1 in /usr/local/lib/python3.7/dist-package
    Requirement already satisfied: prophet in /usr/local/lib/python3.7/dist-packages (1.0.1)
    Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.7/dist-packages (fro
    Requirement already satisfied: Cython!=0.25.1,>=0.22 in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: convertdate>=2.1.2 in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: cmdstanpy==0.9.68 in /usr/local/lib/python3.7/dist-packag
```

```
Requirement already satisfied: setuptools-git>=1.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: matplotlib>=2.0.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: holidays>=0.10.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: LunarCalendar>=0.0.9 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: ujson in /usr/local/lib/python3.7/dist-packages (from cmd
Requirement already satisfied: pymeeus<=1,>=0.3.13 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: hijri-converter in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: korean-lunar-calendar in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from holid
Requirement already satisfied: ephem>=3.7.5.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from Luna
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/li
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
```

```
from prophet import Prophet
m = Prophet()
m.fit(data_raw)
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to o
<prophet.forecaster.Prophet at 0x7f8afbdaf710>
```

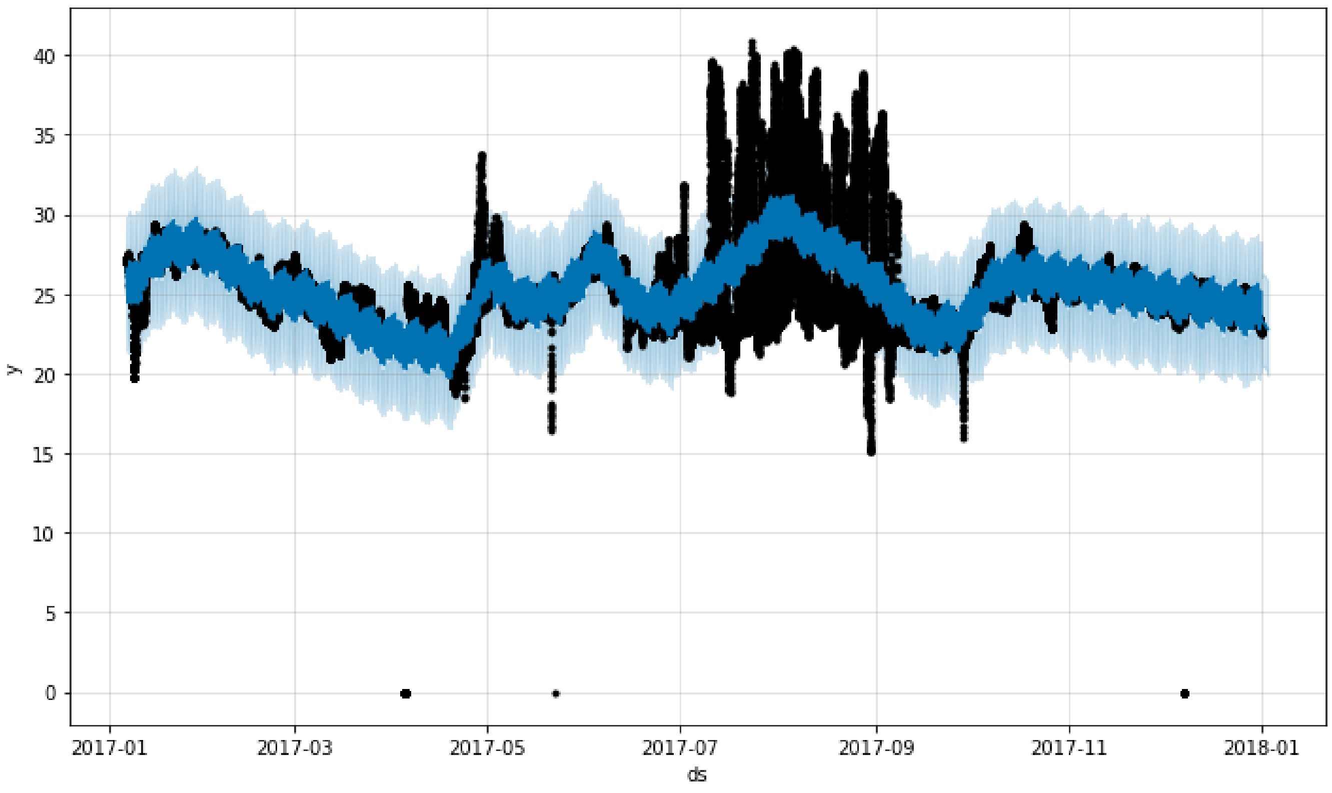
```
future = m.make_future_dataframe(periods=2)
future.tail()
```

	ds
102990	2017-12-31 23:45:00
102991	2017-12-31 23:50:00
102992	2017-12-31 23:55:00
102993	2018-01-01 23:55:00
102994	2018-01-02 23:55:00

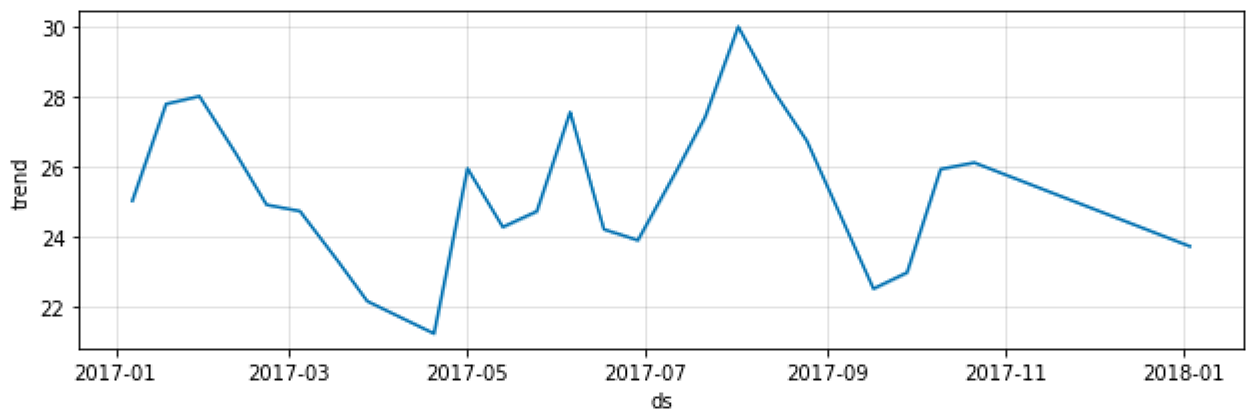
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

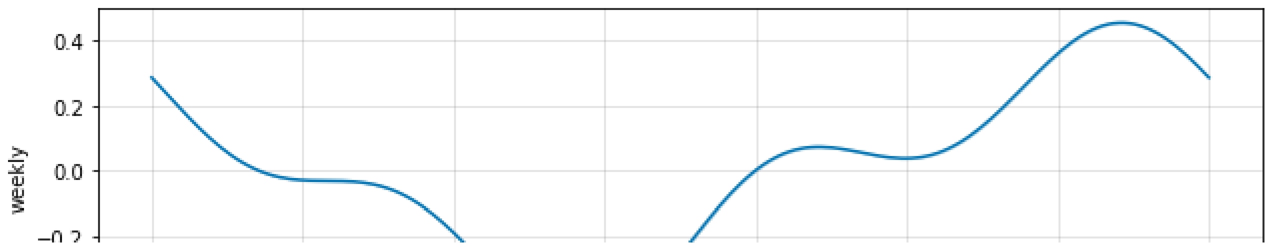
	ds	yhat	yhat_lower	yhat_upper
102990	2017-12-31 23:45:00	23.305936	20.436010	26.041110
102991	2017-12-31 23:50:00	23.295306	20.236410	26.142588

```
fig1 = m.plot(forecast)
```



```
fig2 = m.plot_components(forecast)
```





```
# Python
from prophet.plot import plot_plotly, plot_components_plotly

plot_plotly(m, forecast)
```

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

```
from time import timezone
import datetime
import psycopg2
import pandas as pd
from datetime import datetime, timedelta

CONNECTION = "postgres://tsdbadmin:AVNS_qdv_vgOhhiXe9T2@tsdb-206531ac-sjsu-0c73.a.timescaledb
conn = psycopg2.connect(CONNECTION)

def hello_world():
  conn = psycopg2.connect(CONNECTION)
  cursor = conn.cursor()
  d = (datetime.today() - timedelta(days=370)).strftime('%Y-%m-%d %H:%M:%S')
  preds = []
  for sensor_id in ['sensor_1', 'sensor_2', 'sensor_3', 'sensor_4', 'sensor_5', 'sensor_6', '
    query = f"SELECT time, temp FROM hvac_data WHERE time > '{d}' AND sensor_id='{sensor_id}'
    cursor.execute(query)
    df = pd.DataFrame(cursor.fetchall(), columns=['ds', 'y'])
    df.ds = pd.to_datetime(df.ds).dt.tz_localize(None)
    from prophet import Prophet
    m = Prophet()
    m.fit(df)
    future = m.make_future_dataframe(periods=1)
    print(future.tail())
```

```
      forecast = m.predict(future)
      forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
      temp_mean = 25.165
      tsd = 3.1120

      prediction = "Normal"
      if (forecast[forecast['yhat'] > temp_mean+tsd].shape[0] > 0):
        prediction = "There is a increasing trend, potential breach"
      if(forecast[forecast['yhat'] < temp_mean-tsd].shape[0] > 0):
        prediction = "There is a decreasing trend, potential breach"
      if (forecast[forecast['yhat'] > temp_mean+2*tsd].shape[0] > 0):
        prediction = "Amber Alert: There is a increasing trend, high possibility of a breach"
      if(forecast[forecast['yhat'] < temp_mean-2*tsd].shape[0] > 0):
        prediction = "Amber Alert: There is a decreasing trend, high possibility of a breach"
      if (forecast[forecast['yhat'] > temp_mean+3*tsd].shape[0] > 0):
        prediction = "Red Alert: Fix required"
      if(forecast[forecast['yhat'] < temp_mean-3*tsd].shape[0] > 0):
        prediction = "Red Alert: Fix required"
      preds.append([sensor_id, prediction])
    return preds
  hello_world()
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                         ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                         ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                         ds
1145 2021-05-10 23:40:00
1146 2021-05-10 23:45:00
1147 2021-05-10 23:50:00
1148 2021-05-10 23:55:00
1149 2021-05-11 23:55:00
[['sensor_1', 'Red Alert: Fix required'],
```

```python
import psycopg2

CONNECTION = "postgres://tsdbadmin:AVNS_qdv_vgOhhiXe9T2@tsdb-206531ac-sjsu-0c73.a.timescaledb
conn = psycopg2.connect(CONNECTION)

def hello_world():
  cursor = conn.cursor()
  d = (datetime.today() - timedelta(days=370)).strftime('%Y-%m-%d %H:%M:%S')
  preds = []
  for sensor_id in ['sensor_1', 'sensor_2', 'sensor_3', 'sensor_4', 'sensor_5', 'sensor_6', '
    query = f"SELECT time, temp, sensor_location FROM hvac_data WHERE time > '{d}' AND sensor
    cursor.execute(query)
    df = pd.DataFrame(cursor.fetchall(), columns=['ds', 'y'])
    df.ds = pd.to_datetime(df.ds).dt.tz_localize(None)
    from prophet import Prophet
    m = Prophet()
    m.fit(df)
    future = m.make_future_dataframe(periods=1)
    print(future.tail())
    forecast = m.predict(future)
    forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
    temp_mean = 25.165
    tsd = 3.1120

    prediction = "Normal"
    if (forecast[forecast['yhat'] > temp_mean+tsd].shape[0] > 0):
      prediction = "There is a increasing trend, potential breach"
    if(forecast[forecast['yhat'] < temp_mean-tsd].shape[0] > 0):
      prediction = "There is a decreasing trend, potential breach"
    if (forecast[forecast['yhat'] > temp_mean+2*tsd].shape[0] > 0):
      prediction = "Amber Alert: There is a increasing trend, high possibility of a breach"
    if(forecast[forecast['yhat'] < temp_mean-2*tsd].shape[0] > 0):
      prediction = "Amber Alert: There is a decreasing trend, high possibility of a breach"
    if (forecast[forecast['yhat'] > temp_mean+3*tsd].shape[0] > 0):
      prediction = "Red Alert: Fix required"
    if(forecast[forecast['yhat'] < temp_mean-3*tsd].shape[0] > 0):
      prediction = "Red Alert: Fix required"
    preds.append([(datetime.today()).strftime('%Y-%m-%d %H:%M:%S'), sensor_id, prediction])
    cursor.execute("INSERT INTO hvac_data_prediction (time, sensor_id, prediction) VALUES (%s
    conn.commit()

  return preds
hello_world()
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
```

```
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True t
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True t
                      ds
1143 2021-05-10 23:40:00
1144 2021-05-10 23:45:00
1145 2021-05-10 23:50:00
1146 2021-05-10 23:55:00
1147 2021-05-11 23:55:00
```