


```
import numpy as np
import pandas as pd
```

Data Collection & Processing

```
air_quality_data =pd.read_csv('/content/AirQualityUCI.csv', sep=';', decimal=',')
```

```
air_quality_data.head()
```



	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08
	00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	
	00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	
	00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	
	00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	
	00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	

Next steps:


[Generate code with air_quality_data](#)

[View recommended plots](#)

[New interactive sheet](#)

```
#removing last 2 column
air_quality_data=air_quality_data.iloc[:, :-2]
```

```
air_quality_data.head()
```



	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	


Next steps:

[Generate code with air_quality_data](#)

[View recommended plots](#)

[New interactive sheet](#)

```
air_quality_data.tail()
```



	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(N
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	!
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	!
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	!
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	!
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	!

```
air_quality_data.shape
```



(9471, 15)

```
air_quality_data.loc[9356]
```



9356

Date	04/04/2005
Time	14.00.00
CO(GT)	2.2
PT08.S1(CO)	1071.0
NMHC(GT)	-200.0
C6H6(GT)	11.9
PT08.S2(NMHC)	1047.0
NOx(GT)	265.0
PT08.S3(NOx)	654.0
NO2(GT)	168.0
PT08.S4(NO2)	1129.0
PT08.S5(O3)	816.0
T	28.5
RH	13.1
AH	0.5028

dtype: object

```
#last data point = 9356
air_quality_data= air_quality_data.head(9357)
```

```
air_quality_data.tail()
```



	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	F
9352	04/04/2005	10.00.00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	
9353	04/04/2005	11.00.00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	
9354	04/04/2005	12.00.00	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	
9355	04/04/2005	13.00.00	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	
9356	04/04/2005	14.00.00	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	

```
air_quality_data.shape
```



(9357, 15)

```
air_quality_data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9357 entries, 0 to 9356
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  9357 non-null   object
1   Time                  9357 non-null   object
2   CO(GT)                9357 non-null   float64
3   PT08.S1(CO)           9357 non-null   float64
4   NMHC(GT)              9357 non-null   float64
5   C6H6(GT)              9357 non-null   float64
6   PT08.S2(NMHC)         9357 non-null   float64
7   NOx(GT)               9357 non-null   float64
8   PT08.S3(NOx)          9357 non-null   float64
9   NO2(GT)               9357 non-null   float64
10  PT08.S4(NO2)          9357 non-null   float64
11  PT08.S5(O3)           9357 non-null   float64
```

```

12 T          9357 non-null float64
13 RH         9357 non-null float64
14 AH         9357 non-null float64
dtypes: float64(13), object(2)
memory usage: 1.1+ MB

```

```
air_quality_data.isnull().sum()
```



	0
Date	0
Time	0
CO(GT)	0
PT08.S1(CO)	0
NMHC(GT)	0
C6H6(GT)	0
PT08.S2(NMHC)	0
NOx(GT)	0
PT08.S3(NOx)	0
NO2(GT)	0
PT08.S4(NO2)	0
PT08.S5(O3)	0
T	0
RH	0
AH	0

```
dtype: int64
```

```
air_quality_data.isin([-200]).sum(axis=0)
```



	0
Date	0
Time	0
CO(GT)	1683
PT08.S1(CO)	366
NMHC(GT)	8443
C6H6(GT)	366
PT08.S2(NMHC)	366
NOx(GT)	1639
PT08.S3(NOx)	366
NO2(GT)	1642
PT08.S4(NO2)	366
PT08.S5(O3)	366
T	366
RH	366
AH	366

```
dtype: int64
```

Handling the missing values

```
#-200 is null value
#replacing -200 with NaN
air_quality_data = air_quality_data.replace(to_replace= -200, value=np.NaN)
```

```
air_quality_data.isnull().sum()
```



	0
Date	0
Time	0
CO(GT)	1683
PT08.S1(CO)	366
NMHC(GT)	8443
C6H6(GT)	366
PT08.S2(NMHC)	366
NOx(GT)	1639
PT08.S3(NOx)	366
NO2(GT)	1642
PT08.S4(NO2)	366
PT08.S5(O3)	366
T	366
RH	366
AH	366

dtype: int64

```
air_quality_data.tail()
```



	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	P
9352	04/04/2005	10.00.00	3.1	1314.0	NaN	13.5	1101.0	472.0	539.0	190.0	
9353	04/04/2005	11.00.00	2.4	1163.0	NaN	11.4	1027.0	353.0	604.0	179.0	
9354	04/04/2005	12.00.00	2.4	1142.0	NaN	12.4	1063.0	293.0	603.0	175.0	
9355	04/04/2005	13.00.00	2.1	1003.0	NaN	9.5	961.0	235.0	702.0	156.0	
9356	04/04/2005	14.00.00	2.2	1071.0	NaN	11.9	1047.0	265.0	654.0	168.0	

```
air_quality_data.mean(numeric_only=True)
```



0

CO(GT)	2.152750
PT08.S1(CO)	1099.833166
NMHC(GT)	218.811816
C6H6(GT)	10.083105
PT08.S2(NMHC)	939.153376
NOx(GT)	246.896735
PT08.S3(NOx)	835.493605
NO2(GT)	113.091251
PT08.S4(NO2)	1456.264598
PT08.S5(O3)	1022.906128
T	18.317829
RH	49.234201
AH	1.025530

dtype: float64

Replacing the null values with mean of respective column


```
air_quality_data = air_quality_data.fillna(air_quality_data.mean(numeric_only=True))
```

```
air_quality_data.tail()
```



	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)
9352	04/04/2005	10.00.00	3.1	1314.0	218.811816	13.5	1101.0	472.0	539.0	190.0
9353	04/04/2005	11.00.00	2.4	1163.0	218.811816	11.4	1027.0	353.0	604.0	179.0
9354	04/04/2005	12.00.00	2.4	1142.0	218.811816	12.4	1063.0	293.0	603.0	175.0
9355	04/04/2005	13.00.00	2.1	1003.0	218.811816	9.5	961.0	235.0	702.0	156.0
9356	04/04/2005	14.00.00	2.2	1071.0	218.811816	11.9	1047.0	265.0	654.0	168.0

```
air_quality_data.isnull().sum()
```



	0
Date	0
Time	0
CO(GT)	0
PT08.S1(CO)	0
NMHC(GT)	0
C6H6(GT)	0
PT08.S2(NMHC)	0
NOx(GT)	0
PT08.S3(NOx)	0
NO2(GT)	0
PT08.S4(NO2)	0
PT08.S5(O3)	0
T	0
RH	0
AH	0


dtype: int64

Two tasks can be performed on this data

1. **Forecasting**
2. **Regression**


Forecasting with *FB Prophet* Algorithm

```
#converting date and time to timestamp(YYYY-MM-DD HH:MM:SS)
date_info = pd.to_datetime(air_quality_data['Date'], format='%d/%m/%Y')
print(date_info)
```



```
0      2004-03-10
1      2004-03-10
2      2004-03-10
3      2004-03-10
4      2004-03-10
...
9352   2005-04-04
9353   2005-04-04
9354   2005-04-04
9355   2005-04-04
9356   2005-04-04
Name: Date, Length: 9357, dtype: datetime64[ns]
```

```
time_info = air_quality_data['Time']
print(time_info)
```



```
0      18.00.00
1      19.00.00
2      20.00.00
3      21.00.00
4      22.00.00
...
9352   10.00.00
9353   11.00.00
9354   12.00.00
9355   13.00.00
9356   14.00.00
Name: Time, Length: 9357, dtype: object
```

```
time_info = time_info.apply(lambda x:x. replace('.',':'))
```

```
print(time_info)
```

```
0      18:00:00
1      19:00:00
2      20:00:00
3      21:00:00
4      22:00:00
...
9352    10:00:00
9353    11:00:00
9354    12:00:00
9355    13:00:00
9356    14:00:00
Name: Time, Length: 9357, dtype: object
```

```
print(type(date_info))
```

```
print(type(time_info))
```

```
<class 'pandas.core.series.Series'>
<class 'pandas.core.series.Series'>
```

```
#combining 2 series to a panda dataframe
```

```
date_time = pd.concat([date_info,time_info],axis=1)
```

```
date_time.head()
```

```
0      2004-03-10  18:00:00
1      2004-03-10  19:00:00
2      2004-03-10  20:00:00
3      2004-03-10  21:00:00
4      2004-03-10  22:00:00
```

Next steps:

[Generate code with date_time](#)
[View recommended plots](#)
[New interactive sheet](#)

```
date_time.shape
```

```
(9357, 2)
```

```
#combining date and time column to single column
```

```
date_time['ds'] = date_time['Date'].astype(str)+' '+date_time['Time'].astype(str)
```

```
date_time.head()
```

```
0      2004-03-10  18:00:00  2004-03-10 18:00:00
1      2004-03-10  19:00:00  2004-03-10 19:00:00
2      2004-03-10  20:00:00  2004-03-10 20:00:00
3      2004-03-10  21:00:00  2004-03-10 21:00:00
4      2004-03-10  22:00:00  2004-03-10 22:00:00
```

Next steps:

[Generate code with date_time](#)
[View recommended plots](#)
[New interactive sheet](#)

```
date_time.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9357 entries, 0 to 9356
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0   Date    9357 non-null    datetime64[ns]
 1   Time    9357 non-null    object
 2   ds       9357 non-null    object
dtypes: datetime64[ns](1), object(2)
memory usage: 219.4+ KB

```

```
data = pd.DataFrame()
```

```
data['ds'] = pd.to_datetime(date_time['ds'])
```

```
data.head()
```

```

ds
0  2004-03-10 18:00:00
1  2004-03-10 19:00:00
2  2004-03-10 20:00:00
3  2004-03-10 21:00:00
4  2004-03-10 22:00:00

```

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
data['y'] = air_quality_data['RH']
```

```
data.head()
```

```

ds      y
0  2004-03-10 18:00:00  48.9
1  2004-03-10 19:00:00  47.7
2  2004-03-10 20:00:00  54.0
3  2004-03-10 21:00:00  60.0
4  2004-03-10 22:00:00  59.6

```

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
pip install prophet
```

```

Requirement already satisfied: prophet in /usr/local/lib/python3.11/dist-packages (1.1.6)
Requirement already satisfied: cmdstanpy>=1.0.4 in /usr/local/lib/python3.11/dist-packages (from prophet) (1.2.5)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.11/dist-packages (from prophet) (1.26.4)
Requirement already satisfied: matplotlib>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from prophet) (3.10.0)
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.11/dist-packages (from prophet) (2.2.2)
Requirement already satisfied: holidays<1,>=0.25 in /usr/local/lib/python3.11/dist-packages (from prophet) (0.64)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.11/dist-packages (from prophet) (4.67.1)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.11/dist-packages (from prophet) (6.5)
Requirement already satisfied: stanio<2.0.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from cmdstanpy>=1.0.4) (0.4.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.11/dist-packages (from holidays<1,>=0.25) (2.9.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (4.55.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=2.0.0) (3.2.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.4) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.0.4) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil->holidays) (1.17.0)

```



```
from prophet import Prophet
```

```
#training the Prophet Model
```

```
model = Prophet()
model.fit(data)
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpliloa3ba/zzv5n8u4.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpliloa3ba/lf49wuv1.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'ra
13:50:08 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
13:50:09 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7bc4f09eb690>
```

```
future = model.make_future_dataframe(periods=365, freq='H')
future.tail()
```

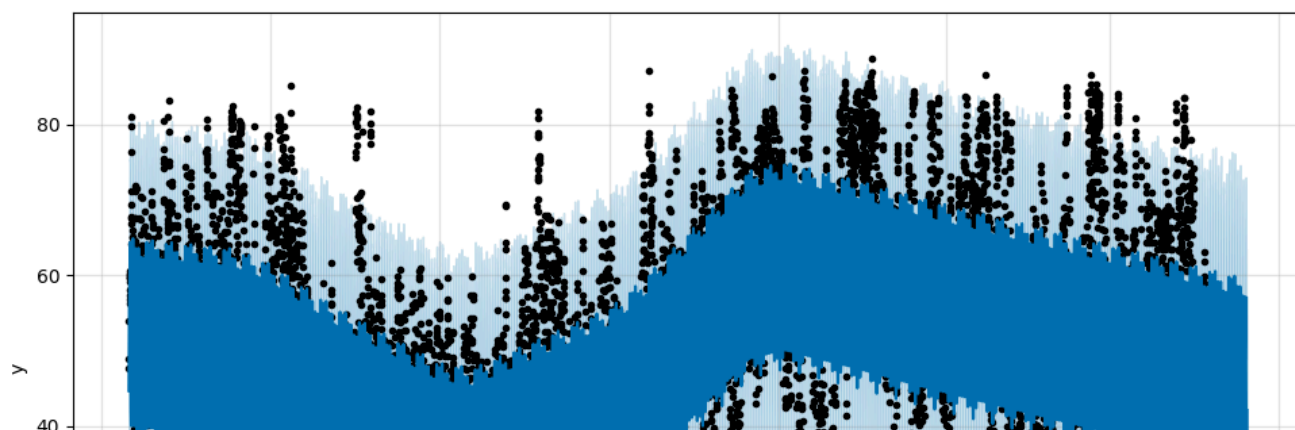
```
/usr/local/lib/python3.11/dist-packages/prophet/forecaster.py:1854: FutureWarning: 'H' is deprecated and will be r
dates = pd.date_range(
```

	ds
9717	2005-04-19 15:00:00
9718	2005-04-19 16:00:00
9719	2005-04-19 17:00:00
9720	2005-04-19 18:00:00
9721	2005-04-19 19:00:00

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

	ds	yhat	yhat_lower	yhat_upper
9717	2005-04-19 15:00:00	33.494204	17.264274	48.984358
9718	2005-04-19 16:00:00	33.977616	17.878340	50.209951
9719	2005-04-19 17:00:00	35.610862	20.751995	50.388344
9720	2005-04-19 18:00:00	38.506470	22.957294	54.214247
9721	2005-04-19 19:00:00	42.060689	25.954014	57.111880

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



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

