```
pip install pmdarima
```

```
Collecting pmdarima
  Downloading pmdarima-1.8.3-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.m
     |████████████████████████████████| 1.4 MB 11.7 MB/s
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pandas>=0.19 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: numpy>=1.19.3 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: Cython!=0.29.18,>=0.29 in /usr/local/lib/python3.7/di
Collecting statsmodels!=0.12.0,>=0.11
  Downloading statsmodels-0.13.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_
     |████████████████████████████████| 9.8 MB 49.1 MB/s
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in /usr/local/lib/python3
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.7/dist-package
Installing collected packages: statsmodels, pmdarima
  Attempting uninstall: statsmodels
    Found existing installation: statsmodels 0.10.2
    Uninstalling statsmodels-0.10.2:
      Successfully uninstalled statsmodels-0.10.2
Successfully installed pmdarima-1.8.3 statsmodels-0.13.0
```

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

Read Data

```
df=pd.read_csv('/content/DATA_SET.csv',index_col='DATE',parse_dates=True)
df=df.dropna()
print('Shape of data',df.shape)
df.head()
```

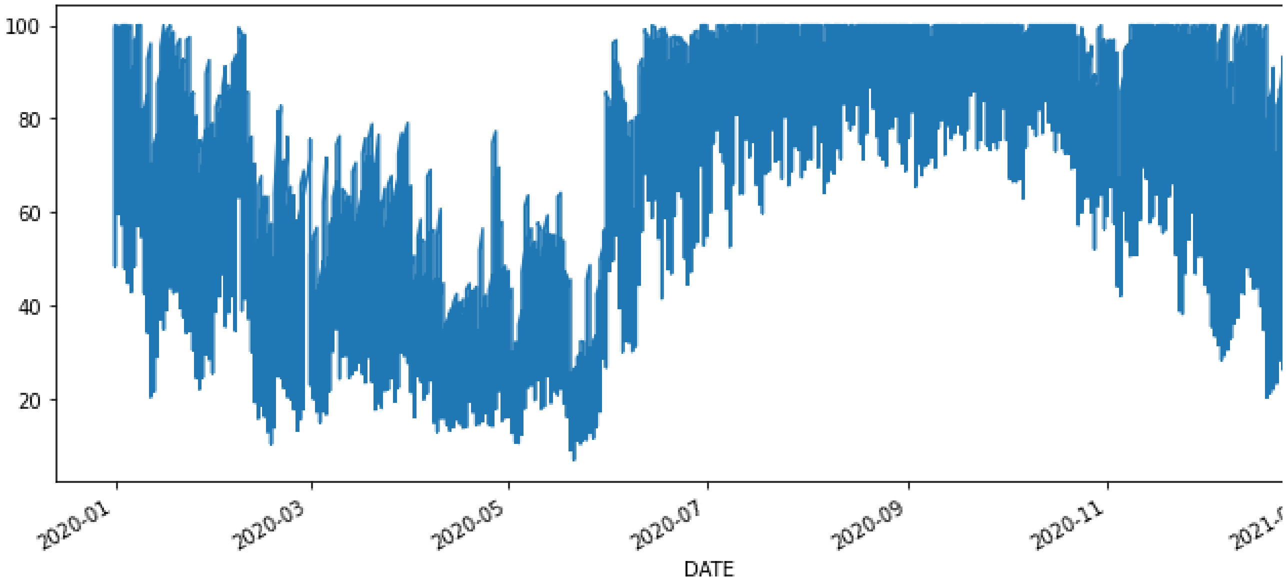
```
Shape of data (8760, 7)
```

	Hour	Solar Zenith Angle	Temperature	DHI	DNI	GHI	Relative Humidity
DATE							
2020-01-01	0	108.31	15.0	0	0	0	100.00
2020-01-01	1	95.08	15.2	0	0	0	100.00
2020-01-01	2	82.19	16.8	22	0	22	96.07
2020-01-01	3	70.16	19.0	153	346	270	86.25
2020-01-01	4	59.21	22.0	43	0	43	68.53

Plot Your Data

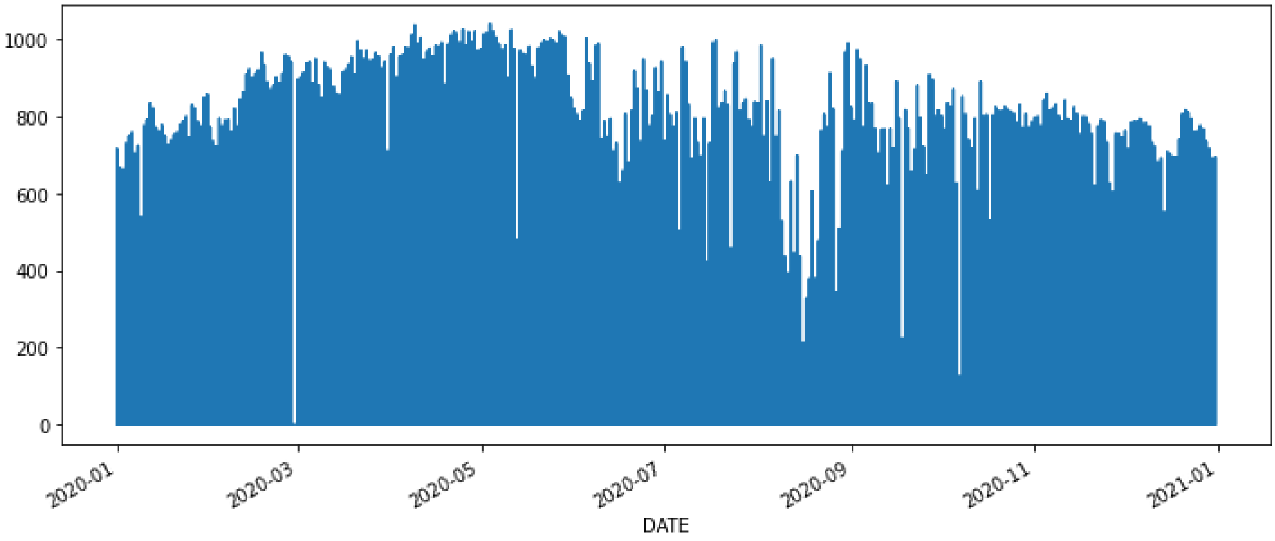
```
df['Relative Humidity'].plot(figsize=(12,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b0aaef2d0>
```



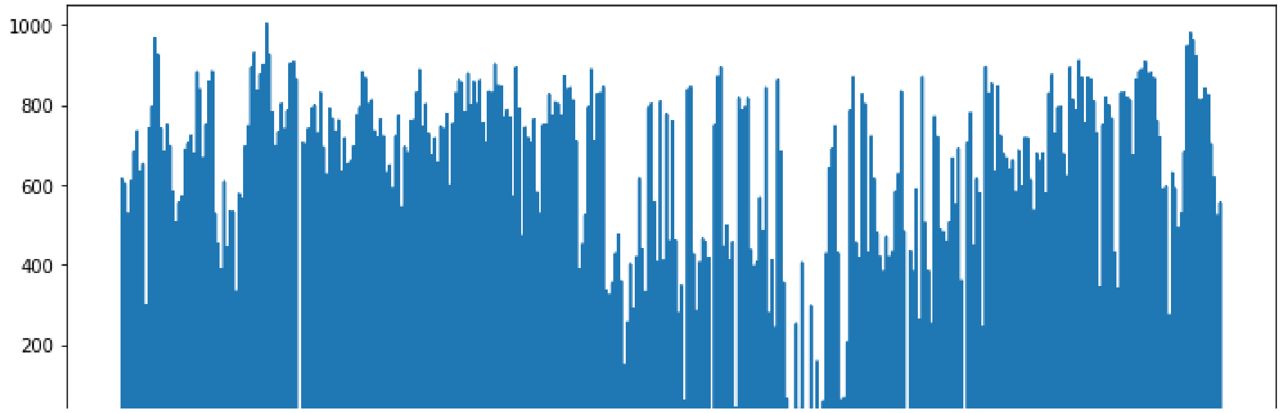
```
df['GHI'].plot(figsize=(12,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b0ac33a90>
```



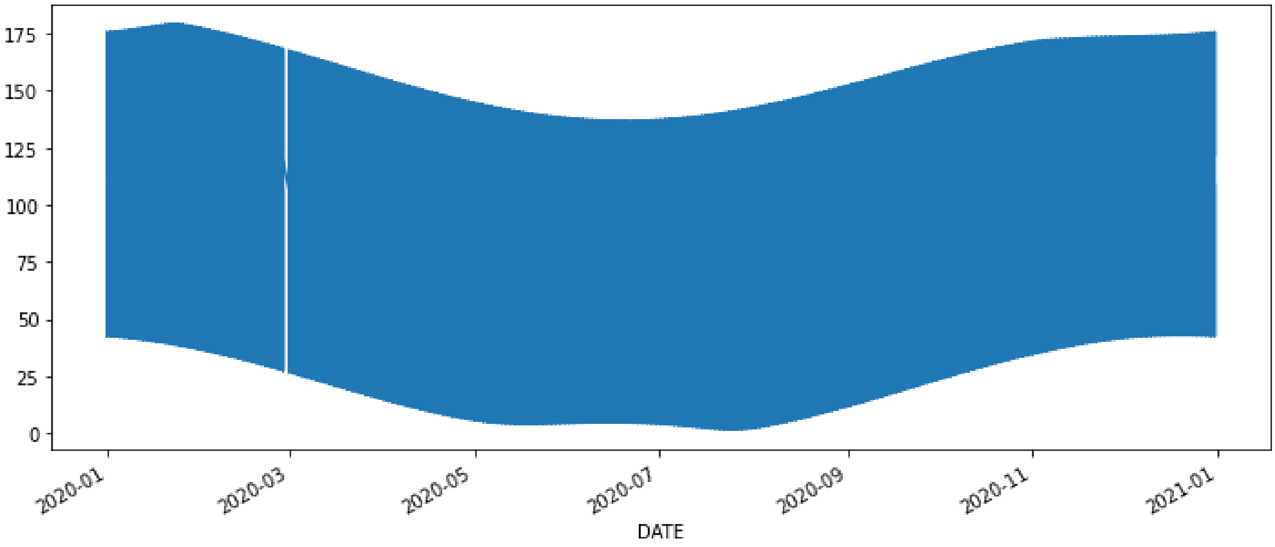
```
df['DNI'].plot(figsize=(12,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b0ac31ed0>
```



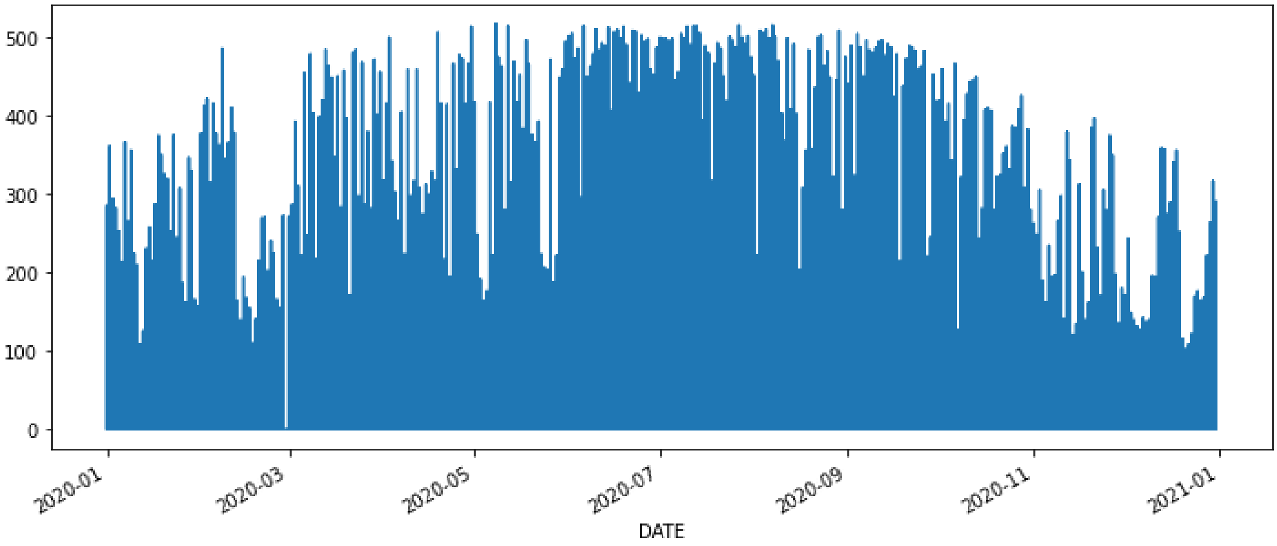
```
df['Solar Zenith Angle'].plot(figsize=(12,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b0b2cc2d0>
```



```
df['DHI'].plot(figsize=(12,5))
```

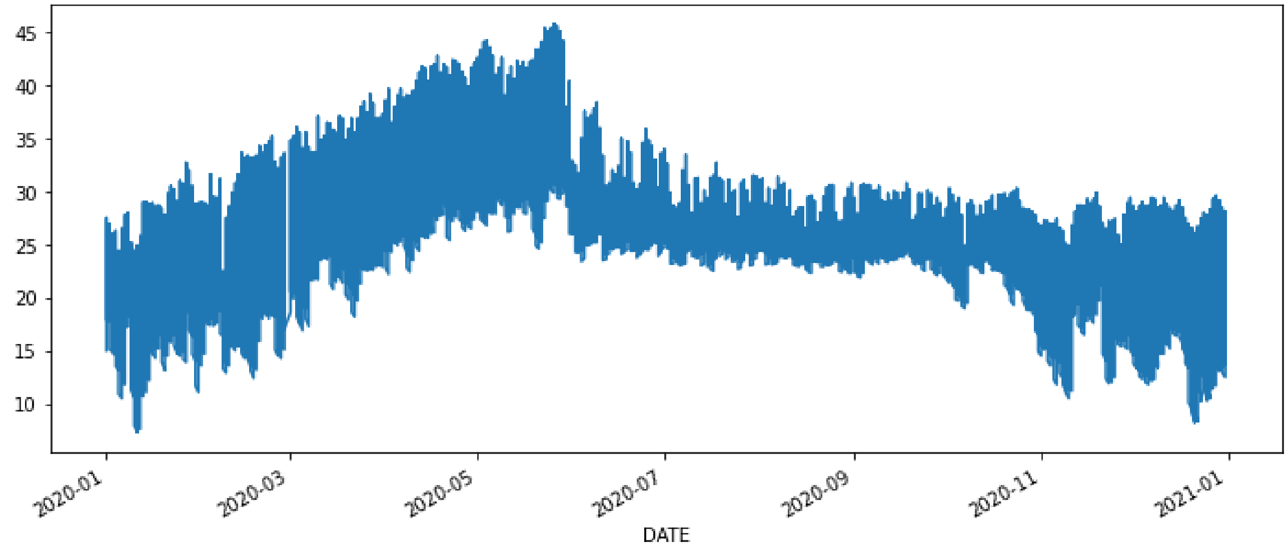
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b0b19c9d0>
```



```
df['Temperature'].plot(figsize=(12,5))
```

```
df['Temperature'].plot(figsize=(12,5))
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7fcbbb66c790>
```



Check For Stationarity

```
from statsmodels.tsa.stattools import adfuller

def adf_test(dataset):
  dftest = adfuller(dataset, autolag = 'AIC')
  print("1. ADF : ",dftest[0])
  print("2. P-Value : ", dftest[1])
  print("3. Num Of Lags : ", dftest[2])
  print("4. Num Of Observations Used For ADF Regression and Critical Values Calculation :'
  print("5. Critical Values :")
  for key, val in dftest[4].items():
      print("\t",key, ": ", val)
```

```
adf_test(df['Temperature'])
```

```
    1. ADF :  -2.4086338534042495
    2. P-Value :  0.1393143810655172
    3. Num Of Lags :  27
    4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 873
    5. Critical Values :
             1% :  -3.4310991094132306
             5% :  -2.8618710565057626
             10% :  -2.5669462164097956
```

Figure Out Order for ARIMA Model

```
from pmdarima import auto_arima
# Ignore harmless warnings
# ignore harmless warnings
import warnings
warnings.filterwarnings("ignore")


stepwise_fit = auto_arima(df['Temperature'],
                          suppress_warnings=True)

stepwise_fit.summary()
```

```
                       SARIMAX Results
   Dep. Variable:     y                  No. Observations: 8760
         Model:       SARIMAX(5, 1, 3)   Log Likelihood    -7844.923
          Date:       Sat, 09 Oct 2021        AIC          15707.845
          Time:       11:34:22                BIC          15771.546
        Sample:       0                       HQIC         15729.550
                      - 8760
Covariance Type: opg
              coef    std err    z       P>|z|  [0.025  0.975]
    ar.L1   0.4492   0.010   46.239    0.000  0.430   0.468
    ar.L2   0.7265   0.009   79.345    0.000  0.709   0.744
    ar.L3   0.2906   0.011   27.300    0.000  0.270   0.311
    ar.L4   -0.8129  0.008   -105.488  0.000  -0.828  -0.798
    ar.L5   0.0861   0.012   7.335     0.000  0.063   0.109
    ma.L1   0.5316   0.004   125.862   0.000  0.523   0.540
    ma.L2   -0.5177  0.004   -126.128  0.000  -0.526  -0.510
    ma.L3   -0.9531  0.004   -226.836  0.000  -0.961  -0.945
   sigma2   0.3509   0.004   96.046    0.000  0.344   0.358
    Ljung-Box (L1) (Q):     0.48  Jarque-Bera (JB):  10097.59
          Prob(Q):          0.49      Prob(JB):      0.00
  Heteroskedasticity (H):   0.70       Skew:         1.10
    Prob(H) (two-sided):    0.00      Kurtosis:      7.78


Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

```
from statsmodels.tsa.arima_model import ARIMA
```

Split Data into Training and Testing

```
print(df.shape)
train=df.iloc[:-30]
test=df.iloc[-30:]
print(train.shape,test.shape)
print(test.iloc[0],test.iloc[-1])
```

```
(8760, 7)
(8730, 7) (30, 7)
Hour                    18.00
```

```
Solar Zenith Angle    168.14
Temperature            15.90
DHI                     0.00
DNI                     0.00
GHI                     0.00
Relative Humidity      80.31
Name: 2020-12-30 00:00:00, dtype: float64 Hour                  23.00
Solar Zenith Angle    121.89
Temperature            13.80
DHI                     0.00
DNI                     0.00
GHI                     0.00
Relative Humidity      89.43
Name: 2020-12-31 00:00:00, dtype: float64
```

Train the Model

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
model=SARIMAX(train['Temperature'],order=(1,0,5))
model=model.fit()
model.summary()
```

SARIMAX Results

Dep. Variable:	Temperature	**No. Observations:**	8730
Model:	SARIMAX(1, 0, 5)	**Log Likelihood**	-8757.992
Date:	Sat, 09 Oct 2021	**AIC**	17529.983
Time:	11:35:34	**BIC**	17579.505
Sample:	0 - 8730	**HQIC**	17546.860
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ar.L1	0.9947	0.001	810.789	0.000	0.992	0.997
ma.L1	1.1472	0.006	177.406	0.000	1.134	1.160
ma.L2	0.9622	0.009	102.274	0.000	0.944	0.981
ma.L3	0.6495	0.013	51.096	0.000	0.625	0.674
ma.L4	0.3388	0.015	22.441	0.000	0.309	0.368
ma.L5	0.1105	0.010	10.881	0.000	0.091	0.130
sigma2	0.4350	0.004	120.373	0.000	0.428	0.442

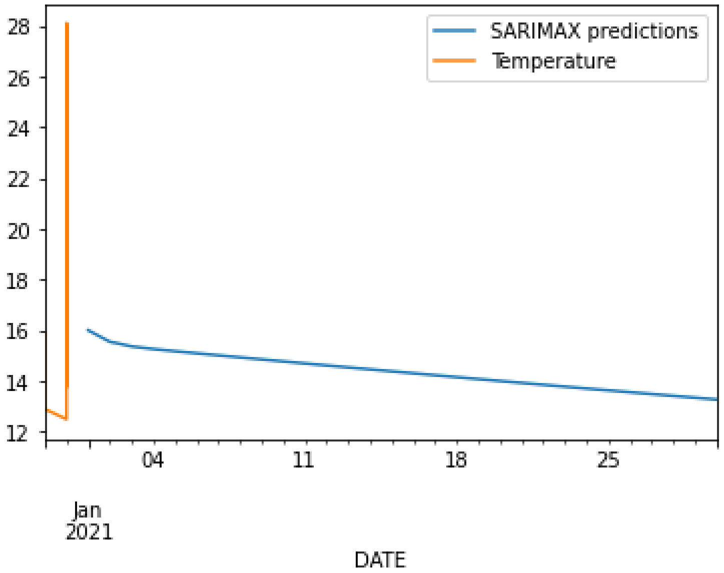
Ljung-Box (L1) (Q):	0.14	**Jarque-Bera (JB):**	11909.48
Prob(Q):	0.71	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.70	**Skew:**	0.94
Prob(H) (two-sided):	0.00	**Kurtosis:**	8.40

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Make Predictions on Test Set

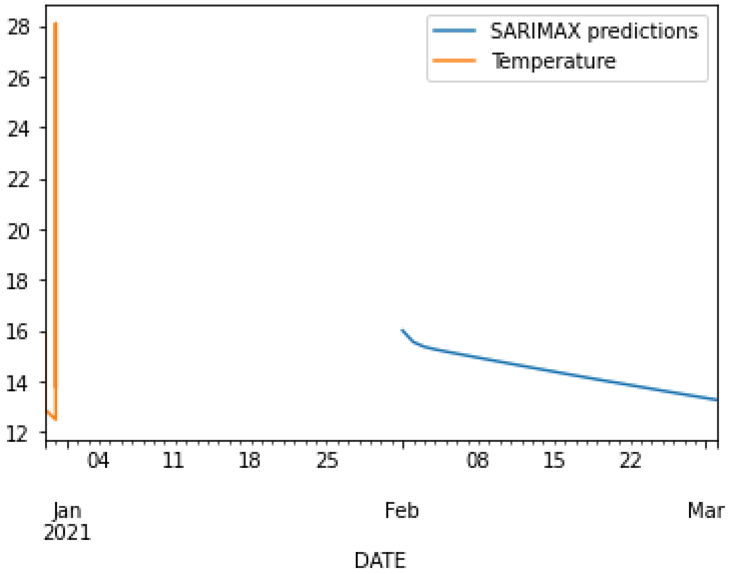
```
start=len(train)
end=len(train)+len(test)-1
#if the predicted values dont have date values as index, you will have to uncomment the fc
index_future_dates=pd.date_range(start='2021-01-01',end='2021-01-30')
pred=model.predict(start=start,end=end,typ='levels').rename('SARIMAX predictions')
pred.index=index_future_dates
pred.plot(legend=True)
test['Temperature'].plot(legend=True)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f9ae13ea410>
```



```
pred.plot(legend='ARIMA Predictions')
test['Temperature'].plot(legend=True)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f9ae104cb90>
```



```
test['Temperature'].mean()
```

```
    18.43
```

```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse=sqrt(mean_squared_error(pred,test['Temperature']))
print(rmse)
```

```
6.634833760407843
```

```
model2=SARIMAX(df['Temperature'],order=(1,0,5))
model2=model2.fit()
df.tail()
```

DATE	Hour	Solar Zenith Angle	Temperature	DHI	DNI	GHI	Relative Humidity
2020-12-31	19	175.67	15.3	0	0	0	79.09
2020-12-31	20	163.12	14.8	0	0	0	82.65
2020-12-31	21	149.37	14.4	0	0	0	85.47
2020-12-31	22	135.58	14.1	0	0	0	87.37
2020-12-31	23	121.89	13.8	0	0	0	89.43

For Future Dates

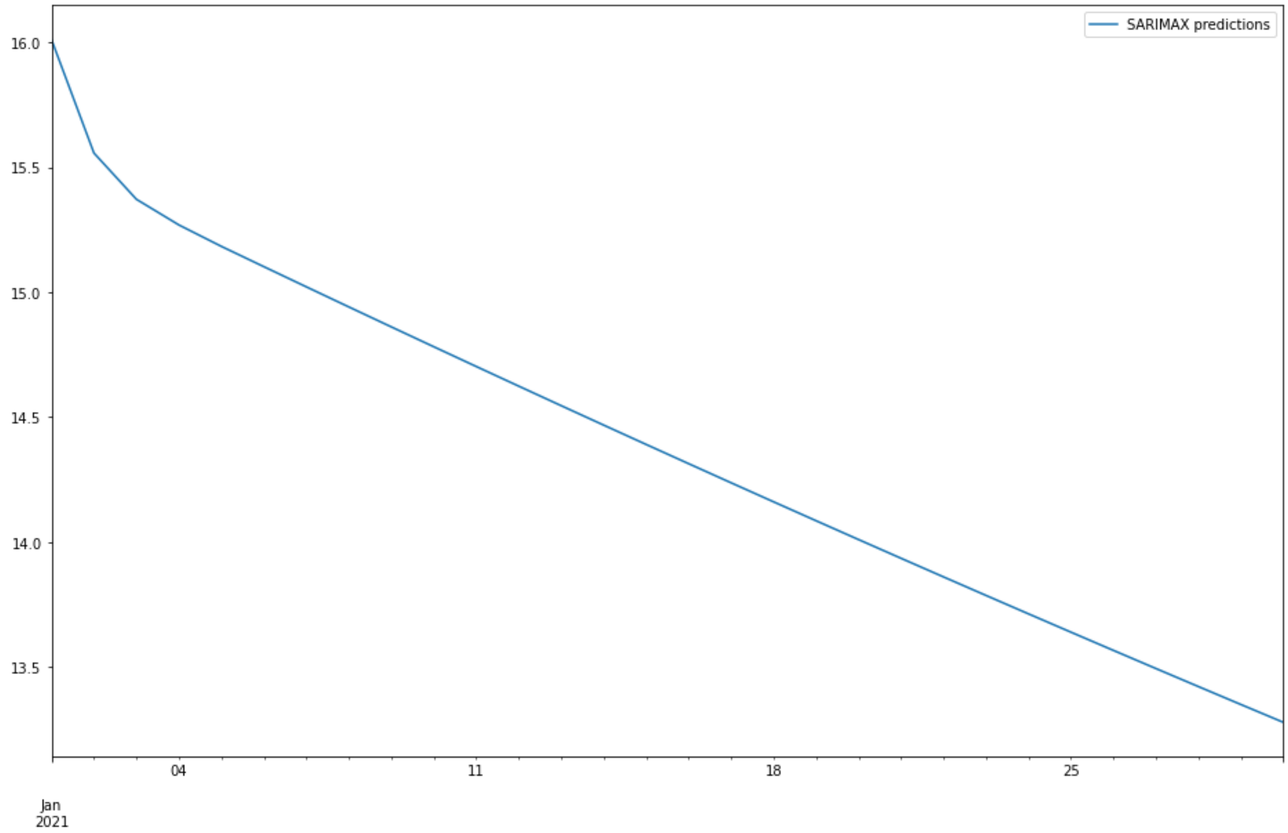
```
index_future_dates=pd.date_range(start='2021-01-01',end='2021-01-31')
#print(index_future_dates)
pred=model2.predict(start=len(df),end=len(df)+30,typ='levels').rename('ARIMA Predictions')
#print(comp_pred)
pred.index=index_future_dates
print(pred)
```

```
2021-01-01    13.564415
2021-01-02    13.393515
2021-01-03    13.266512
2021-01-04    13.166483
2021-01-05    13.086262
2021-01-06    13.016073
2021-01-07    12.946261
2021-01-08    12.876822
2021-01-09    12.807756
2021-01-10    12.739061
2021-01-11    12.670734
2021-01-12    12.602774
2021-01-13    12.535178
2021-01-14    12.467944
2021-01-15    12.401072
2021-01-16    12.334558
2021-01-17    12.268400
2021-01-18    12.202598
2021-01-19    12.137148
2021-01-20    12.072050
```

```
2021-01-21    12.007300
2021-01-22    11.942898
2021-01-23    11.878842
2021-01-24    11.815129
2021-01-25    11.751757
2021-01-26    11.688726
2021-01-27    11.626033
2021-01-28    11.563676
2021-01-29    11.501653
2021-01-30    11.439963
2021-01-31    11.378604
Freq: D, Name: ARIMA Predictions, dtype: float64
```

```
pred.plot(figsize=(16,10),legend=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9ae10f19d0>
```



● ✕