In [1]:
```
from matplotlib import pyplot as plt
from matplotlib.dates import MonthLocator, num2date
from matplotlib.ticker import FuncFormatter
from prophet import Prophet
from prophet.diagnostics import cross_validation, performance_metrics
from prophet.plot import add_changepoints_to_plot

import pandas as pd
import numpy as np
import datetime as dt
from collections import defaultdict
import time
import datetime as dt
from pytz import timezone
tz = timezone('EST')
from tqdm import tqdm

from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error,

import seaborn as sns
%config InlineBackend.figure_format = 'retina'
%matplotlib inline
from matplotlib import pyplot as plt
from matplotlib import style
sns.set()
```

In [2]:
```
ct2011 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2011 = pd.read_excel(ct2011, 'CT')
ct2012 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2012 = pd.read_excel(ct2012, 'CT')
ct2013 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2013 = pd.read_excel(ct2013, 'CT')
ct2014 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2014 = pd.read_excel(ct2014, 'CT')
ct2015 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2015 = pd.read_excel(ct2015, 'CT')
ct2016 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ct2016 = pd.read_excel(ct2016, 'CT')
```

In [3]:
```
ct2017 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ct2018 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ct2019 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ct2020 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ct2021 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ct2022 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
```

In [4]:
```
ct2022
```

Out[4]:
```
           Date  Hr_End  DA_Demand  RT_Demand  DA_LMP  DA_EC  DA_CC  DA_MLC  RT_LMP  RT_
  0  2022-01-01       1     2675.9   2461.367   30.71  32.35  -0.54   -1.10   25.26   25.
  1  2022-01-01       2     2570.2   2336.520   30.45  32.31  -0.67   -1.19   25.11   25.
  2  2022-01-01       3     2397.1   2247.378   29.67  30.85   0.00   -1.18   26.65   27.
  3  2022-01-01       4     2332.6   2192.153   28.59  29.69   0.00   -1.10   24.34   25.
  4  2022-01-01       5     2316.5   2179.424   29.66  30.86   0.00   -1.20   28.30   29.
...         ...     ...        ...        ...     ...    ...    ...     ...     ...
739  2022-01-31      20     4377.4   4402.159  223.75 226.40   0.00   -2.65  291.87  295.
740  2022-01-31      21     4239.4   4264.642  200.92 202.82   0.00   -1.90  261.21  264.
741  2022-01-31      22     3973.5   4067.704  182.19 183.42   0.00   -1.23  249.44  251.
742  2022-01-31      23     3702.2   3787.636  178.46 179.93   0.00   -1.47  190.19  190.
743  2022-01-31      24     3398.2   3569.461  188.26 190.54   0.00   -2.28  188.69  189.

744 rows × 14 columns
```

In [5]:
```python
val2011 = ct2011['DEMAND']
val2012 = ct2012['DEMAND']
val2013 = ct2013['DEMAND']
val2014 = ct2014['DEMAND']
val2015 = ct2015['DEMAND']
val2016 = ct2016['RT_Demand']
val2017 = ct2017['RT_Demand']
val2018 = ct2018['RT_Demand']
val2019 = ct2019['RT_Demand']
val2020 = ct2020['RT_Demand']
val2021 = ct2021['RT_Demand']
val2022 = ct2022['RT_Demand']
```

In [6]:
```python
values = [val2011, val2012, val2013, val2014, val2015, val2016, val2017, val2018,
values_df = pd.concat(values, axis=0, ignore_index=False)
values_df = values_df.reset_index()
period = len(values_df)
```

In [7]:
```
rng = pd.date_range('2011-01-01', periods=period, freq='1H')
date_df = pd.DataFrame({ 'ds': rng})
date_df = date_df.reset_index()
```

In [8]:
```
frames = [date_df, values_df]
ct_load = pd.concat(frames, axis=1, ignore_index=False)
ct_load = ct_load.rename(columns={ct_load.columns[1]: 'ds', ct_load.columns[3]: '
frames2 = [ct_load['ds'], ct_load['y']]
ct_load = pd.concat(frames2, axis=1, ignore_index=False)
ct_load
```

Out[8]:

	ds	y
0	2011-01-01 00:00:00	3053.000
1	2011-01-01 01:00:00	2892.000
2	2011-01-01 02:00:00	2774.000
3	2011-01-01 03:00:00	2710.000
4	2011-01-01 04:00:00	2698.000
...	...	...
97171	2022-01-31 19:00:00	4402.159
97172	2022-01-31 20:00:00	4264.642
97173	2022-01-31 21:00:00	4067.704
97174	2022-01-31 22:00:00	3787.636
97175	2022-01-31 23:00:00	3569.461

97176 rows × 2 columns

In [9]:
```
model = Prophet(
        changepoint_prior_scale=0.5,
        seasonality_mode='multiplicative',
        interval_width=0.95,
    )
model.add_country_holidays(country_name='US')
```

Out[9]:
```
<prophet.forecaster.Prophet at 0x210f8900940>
```

In [10]:
```
model.fit(ct_load)
```

Out[10]:
```
<prophet.forecaster.Prophet at 0x210f8900940>
```

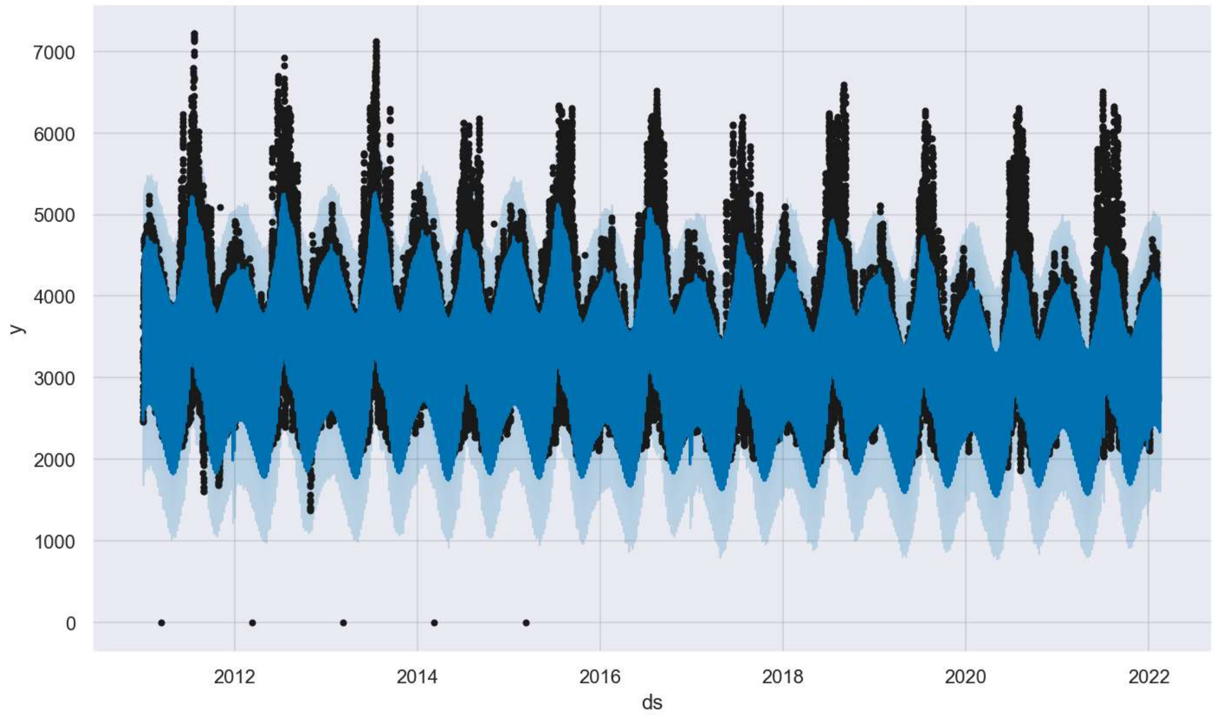
In [11]:
```
future_pd = model.make_future_dataframe(
        periods=365,
        freq='1H',
        include_history=True
    )

# make predictions
forecast_pd = model.predict(future_pd)
```

In [12]:
```
forecast_pd[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

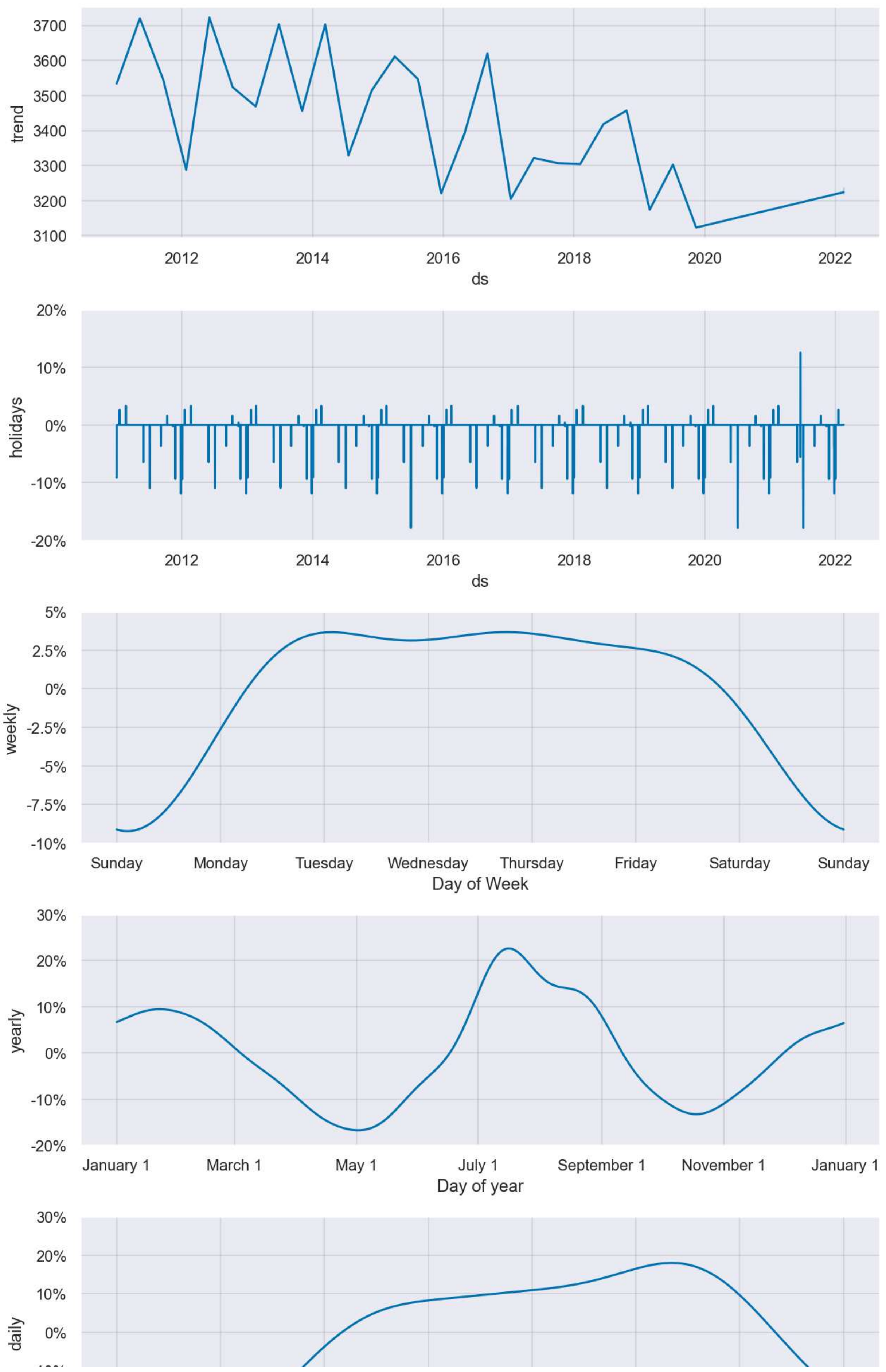
Out[12]:

	ds	yhat	yhat_lower	yhat_upper
97536	2022-02-16 00:00:00	2965.697944	2178.989255	3683.508085
97537	2022-02-16 01:00:00	2814.227948	2070.661236	3554.947083
97538	2022-02-16 02:00:00	2724.002501	1927.141231	3466.096440
97539	2022-02-16 03:00:00	2706.623871	1969.945282	3552.720540
97540	2022-02-16 04:00:00	2781.840752	1940.953899	3502.704911

In [13]:
```
fig1 = model.plot(forecast_pd)
```



```
In [14]: fig2 = model.plot_components(forecast_pd)
```



-10%
-20%
-30%
00:00:00
03:25:42
06:51:25
10:17:08
13:42:51
17:08:34
20:34:17
00:00:00
Hour of day

In [15]:
```
forecast_pd
```

Out[15]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	Christmas Day	Chr Day
0	2011-01-01 00:00:00	3533.640238	2038.874394	3575.975500	3533.640238	3533.640238	0.0	
1	2011-01-01 01:00:00	3533.700184	1878.305674	3422.933806	3533.700184	3533.700184	0.0	
2	2011-01-01 02:00:00	3533.760130	1739.572143	3274.661429	3533.760130	3533.760130	0.0	
3	2011-01-01 03:00:00	3533.820076	1679.270879	3180.964512	3533.820076	3533.820076	0.0	
4	2011-01-01 04:00:00	3533.880021	1828.270751	3269.254868	3533.880021	3533.880021	0.0	
...	...	...	...	...	...	...	...	
97536	2022-02-16 00:00:00	3224.075259	2178.989255	3683.508085	3218.849767	3237.176526	0.0	
97537	2022-02-16 01:00:00	3224.080356	2070.661236	3554.947083	3218.773419	3237.271148	0.0	
97538	2022-02-16 02:00:00	3224.085453	1927.141231	3466.096440	3218.697072	3237.365771	0.0	
97539	2022-02-16 03:00:00	3224.090551	1969.945282	3552.720540	3218.620724	3237.460393	0.0	
97540	2022-02-16 04:00:00	3224.095648	1940.953899	3502.704911	3218.544377	3237.555015	0.0	

97541 rows × 73 columns

In [ ]: