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]:
```
ri2011 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2011 = pd.read_excel(ri2011, 'RI')
ri2012 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2012 = pd.read_excel(ri2012, 'RI')
ri2013 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2013 = pd.read_excel(ri2013, 'RI')
ri2014 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2014 = pd.read_excel(ri2014, 'RI')
ri2015 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2015 = pd.read_excel(ri2015, 'RI')
ri2016 = pd.ExcelFile(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_2011
ri2016 = pd.read_excel(ri2016, 'RI')
```

In [3]:
```
ri2017 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ri2018 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ri2019 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ri2020 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ri2021 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
ri2022 = pd.read_excel(r"C:\Users\Rohan\Desktop\Big Data\Load Data\smd_hourly_201
```

In [4]:
```
ri2022
```

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	699.2	693.413	32.42	32.35	0.23	-0.16	25.83	25.
1	2022-01-01	2	677.8	662.089	32.54	32.31	0.28	-0.05	25.87	25.
2	2022-01-01	3	645.2	638.986	30.74	30.85	0.00	-0.11	27.53	27.
3	2022-01-01	4	642.4	625.841	29.59	29.69	0.00	-0.10	25.14	25.
4	2022-01-01	5	637.7	623.152	30.74	30.86	0.00	-0.12	29.26	29.
...	...	...	...	...	...	...	...	...	...	...
739	2022-01-31	20	1118.3	1141.213	227.10	226.40	0.00	0.70	296.70	295.
740	2022-01-31	21	1064.9	1101.191	203.17	202.82	0.00	0.35	264.93	264.
741	2022-01-31	22	985.5	1046.707	183.95	183.42	0.00	0.53	252.53	251.
742	2022-01-31	23	907.8	984.058	179.94	179.93	0.00	0.01	191.17	190.
743	2022-01-31	24	842.3	928.069	192.34	190.54	0.00	1.80	189.69	189.

744 rows × 14 columns

In [5]:
```
val2011 = ri2011['DEMAND']
val2012 = ri2012['DEMAND']
val2013 = ri2013['DEMAND']
val2014 = ri2014['DEMAND']
val2015 = ri2015['DEMAND']
val2016 = ri2016['RT_Demand']
val2017 = ri2017['RT_Demand']
val2018 = ri2018['RT_Demand']
val2019 = ri2019['RT_Demand']
val2020 = ri2020['RT_Demand']
val2021 = ri2021['RT_Demand']
val2022 = ri2022['RT_Demand']
```

In [6]:
```
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]
ri_load = pd.concat(frames, axis=1, ignore_index=False)
ri_load = ri_load.rename(columns={ri_load.columns[1]: 'ds', ri_load.columns[3]: '
frames2 = [ri_load['ds'], ri_load['y']]
ri_load = pd.concat(frames2, axis=1, ignore_index=False)
ri_load
```

Out[8]:

	ds	y
0	2011-01-01 00:00:00	775.000
1	2011-01-01 01:00:00	733.000
2	2011-01-01 02:00:00	702.000
3	2011-01-01 03:00:00	684.000
4	2011-01-01 04:00:00	681.000
...	...	...
97171	2022-01-31 19:00:00	1141.213
97172	2022-01-31 20:00:00	1101.191
97173	2022-01-31 21:00:00	1046.707
97174	2022-01-31 22:00:00	984.058
97175	2022-01-31 23:00:00	928.069

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 0x172218b5d60>
```

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

Out[10]:
```
<prophet.forecaster.Prophet at 0x172218b5d60>
```

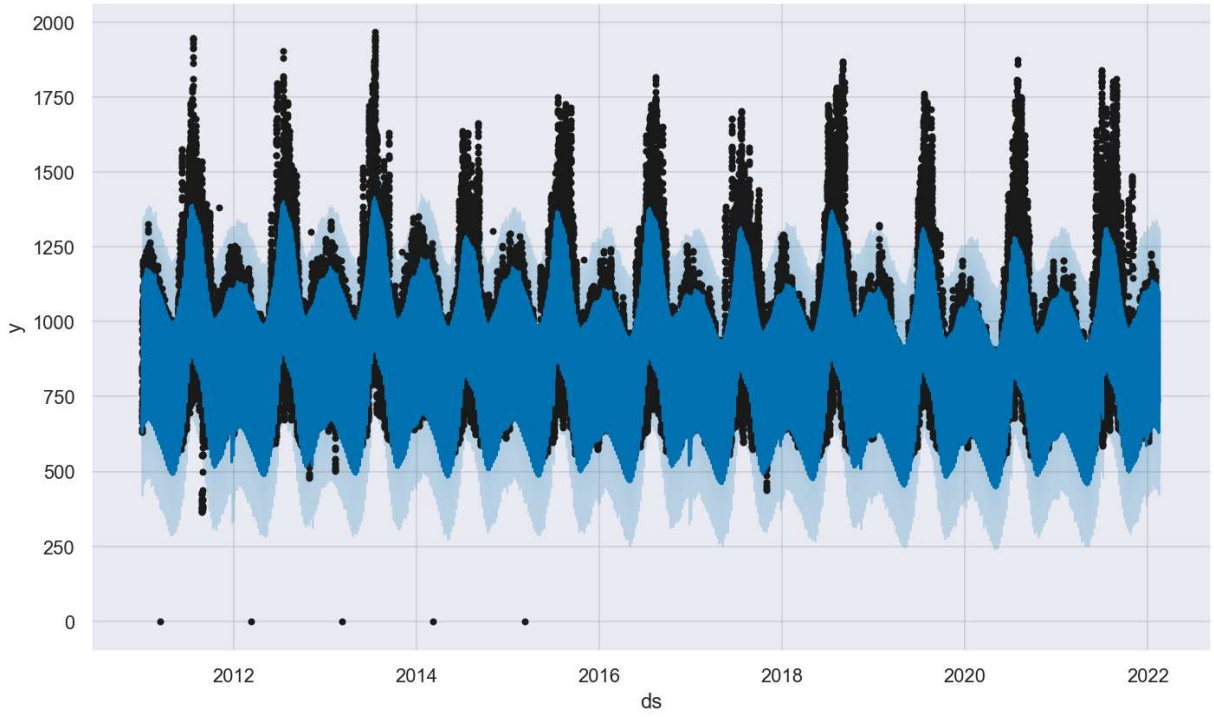
In [11]:
```
future_pd = model.make_future_dataframe(
        periods=365,
        freq='1H',
        include_history=True
    )

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

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

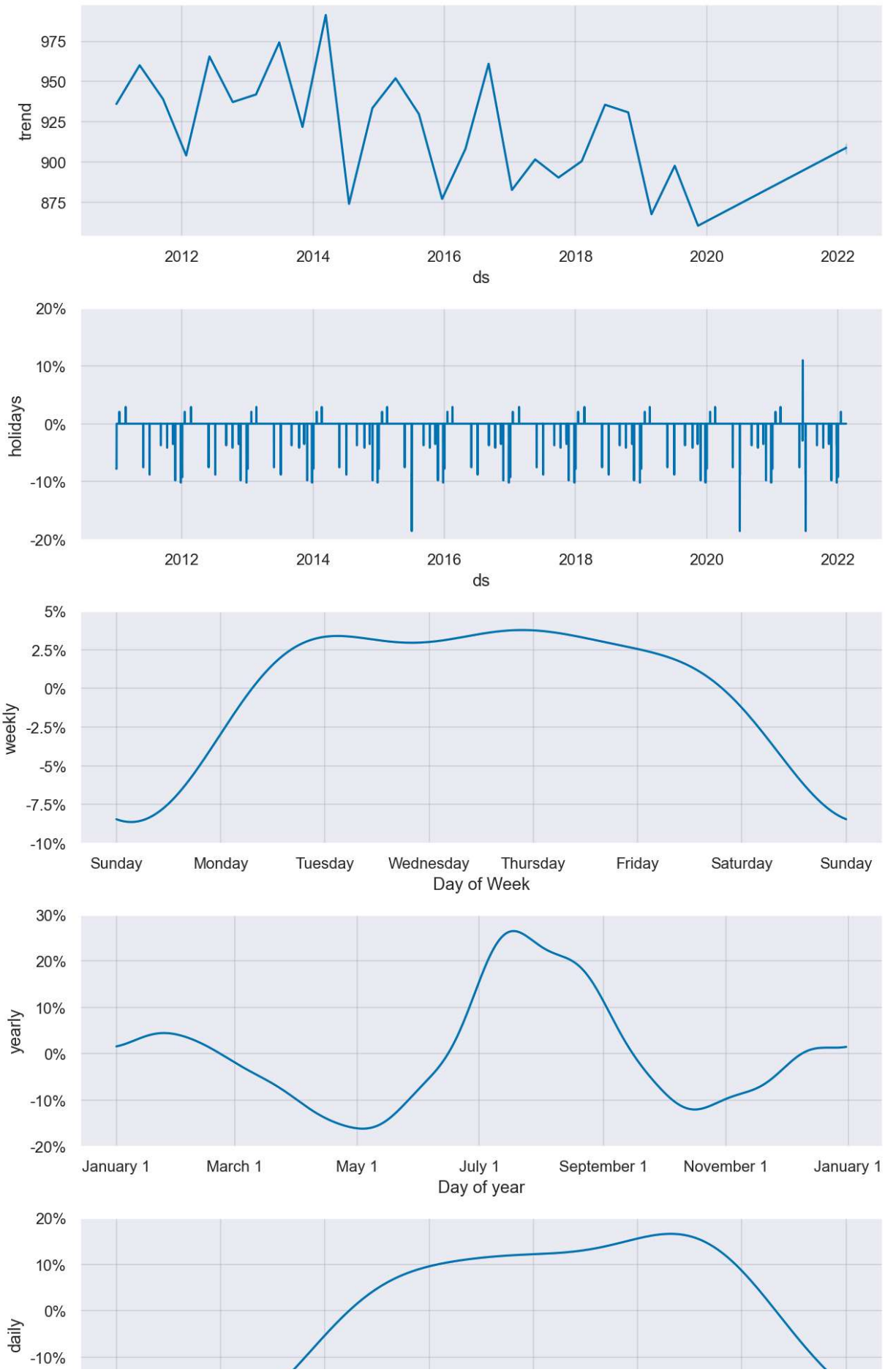
Out[12]:

	ds	yhat	yhat_lower	yhat_upper
97536	2022-02-16 00:00:00	798.123495	599.344577	993.996024
97537	2022-02-16 01:00:00	758.502063	565.280485	955.048214
97538	2022-02-16 02:00:00	735.267810	546.581224	927.245388
97539	2022-02-16 03:00:00	729.627914	529.106838	923.380495
97540	2022-02-16 04:00:00	745.894001	527.262747	945.770642

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



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



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

Out[15]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	Christmas Day	Christm Day_lov
0	2011-01-01 00:00:00	936.122990	499.060706	907.386175	936.122990	936.122990	0.0	
1	2011-01-01 01:00:00	936.130718	476.948914	868.036109	936.130718	936.130718	0.0	
2	2011-01-01 02:00:00	936.138447	421.885007	830.431302	936.138447	936.138447	0.0	
3	2011-01-01 03:00:00	936.146175	429.554392	820.737068	936.146175	936.146175	0.0	
4	2011-01-01 04:00:00	936.153904	440.630758	840.178129	936.153904	936.153904	0.0	
...	...	...	...	...	...	...	...	
97536	2022-02-16 00:00:00	908.956471	599.344577	993.996024	905.589641	911.498232	0.0	
97537	2022-02-16 01:00:00	908.958911	565.280485	955.048214	905.576123	911.511475	0.0	
97538	2022-02-16 02:00:00	908.961350	546.581224	927.245388	905.562606	911.524718	0.0	
97539	2022-02-16 03:00:00	908.963790	529.106838	923.380495	905.549088	911.537961	0.0	
97540	2022-02-16 04:00:00	908.966230	527.262747	945.770642	905.535571	911.551204	0.0	

97541 rows × 73 columns

In [ ]:
```
```