```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
data=pd.read_csv('./data/household_power_consumption.txt',sep=';', header=0, low_memory=Fa
                 parse_dates={'datetime':[0,1]}, index_col=['datetime'])
data.head()
```

datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub
2006-12-16 17:24:00	4.216	0.418	234.840	18.400	
2006-12-16 17:25:00	5.360	0.436	233.630	23.000	
2006-12-16	5.374	0.498	233.290	23.000	

```
data.replace('?',np.nan, inplace=True)
data=data.astype('float32')
```

```
data.isnull().sum()
```

```
Global_active_power      25979
Global_reactive_power    25979
Voltage                  25979
Global_intensity         25979
Sub_metering_1           25979
Sub_metering_2           25979
Sub_metering_3           25979
dtype: int64
```

```
data=data.fillna(data.mean())
data.isnull().sum()
```

```
Global_active_power      0
Global_reactive_power    0
Voltage                  0
Global_intensity         0
Sub_metering_1           0
Sub_metering_2           0
Sub_metering_3           0
dtype: int64
```

```
data.describe()
```

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	S
count	2.075259e+06	2.075259e+06	2.075259e+06	2.075259e+06	
mean	1.091373e+00	1.237164e-01	2.435335e+02	4.630594e+00	
std	1.050167e+00	1.121103e-01	4.183342e+00	4.412327e+00	
min	7.600000e-02	0.000000e+00	2.232000e+02	2.000000e-01	
25%	3.100000e-01	4.800000e-02	2.390200e+02	1.400000e+00	
50%	6.300000e-01	1.020000e-01	2.410500e+02	2.800000e+00	
75%	1.520000e+00	1.920000e-01	2.429700e+02	6.400000e+00	

```
data.corr()
```

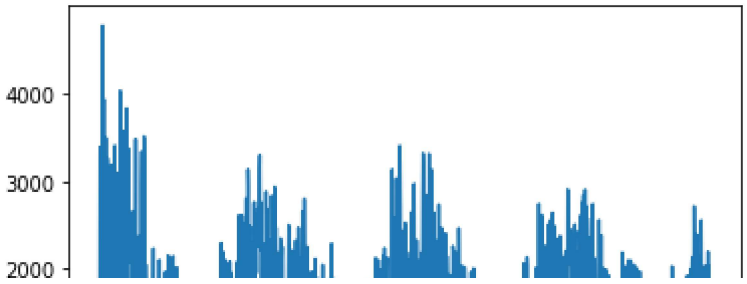
	Global_active_power	Global_reactive_power	Voltage	Global_
Global_active_power	1.000000	0.247017	-0.398231	
Global_reactive_power	0.247017	1.000000	-0.111817	
Voltage	-0.398231	-0.111817	1.000000	
Global_intensity	0.998889	0.266120	-0.409785	
Sub_metering_1	0.484401	0.123111	-0.195225	
Sub_metering_2	0.434569	0.139231	-0.166764	
Sub_metering_3	0.638555	0.089617	-0.267145	

```
resampled_data = data.resample('D').sum()
resampled_data = resampled_data.reset_index()
resampled_data.head()
```

	datetime	Global_active_power	Global_reactive_power	Voltage	Global_intens
0	2006-12-16	1209.176025	34.922001	93552.53125	5180.799
1	2006-12-17	3390.459961	226.005997	345725.31250	14398.599
2	2006-12-18	2203.825928	161.792007	347373.62500	9247.200
3	2006-12-				

```
plt.plot(resampled_data.index,resampled_data.Global_active_power)
```

```
[<matplotlib.lines.Line2D at 0x7f677aa8b290>]
```




```
resampled_data.corr()
```

	Global_active_power	Global_reactive_power	Voltage	Global_
Global_active_power	1.000000	0.041098	0.062393	
Global_reactive_power	0.041098	1.000000	0.050946	
Voltage	0.062393	0.050946	1.000000	
Global_intensity	0.999182	0.062937	0.052214	
Sub_metering_1	0.545338	0.319394	-0.003525	
Sub_metering_2	0.481370	0.182199	-0.013305	
Sub_metering_3	0.732613	0.035568	0.095189	

```
from scipy.stats import pearsonr
corr,_= pearsonr(resampled_data['Global_active_power'],resampled_data['Global_reactive_pow
corr
```

```
0.04109788470945897
```

```
corr2,_ = pearsonr(resampled_data['Voltage'],resampled_data['Global_intensity'])
corr2
```

```
0.05221350712869457
```

```
columns={"datetime": "ds", "Global_active_power": "y"}
X_train = resampled_data[:-365]
X_test = resampled_data[-365:]
X_train = X_train.rename(columns=columns)
X_test = X_test.rename(columns=columns)
```

```
X_train=X_train.iloc[:,:2]
X_train.head()
```

	ds	y
0	2006-12-16	1209.176025

```
X_test=X_test.iloc[:,:2]
X_test.head()
```

	ds	y
1077	2009-11-27	1380.026001
1078	2009-11-28	1858.949951
1079	2009-11-29	1650.962036
1080	2009-11-30	1745.189941
1081	2009-12-01	1756.378052

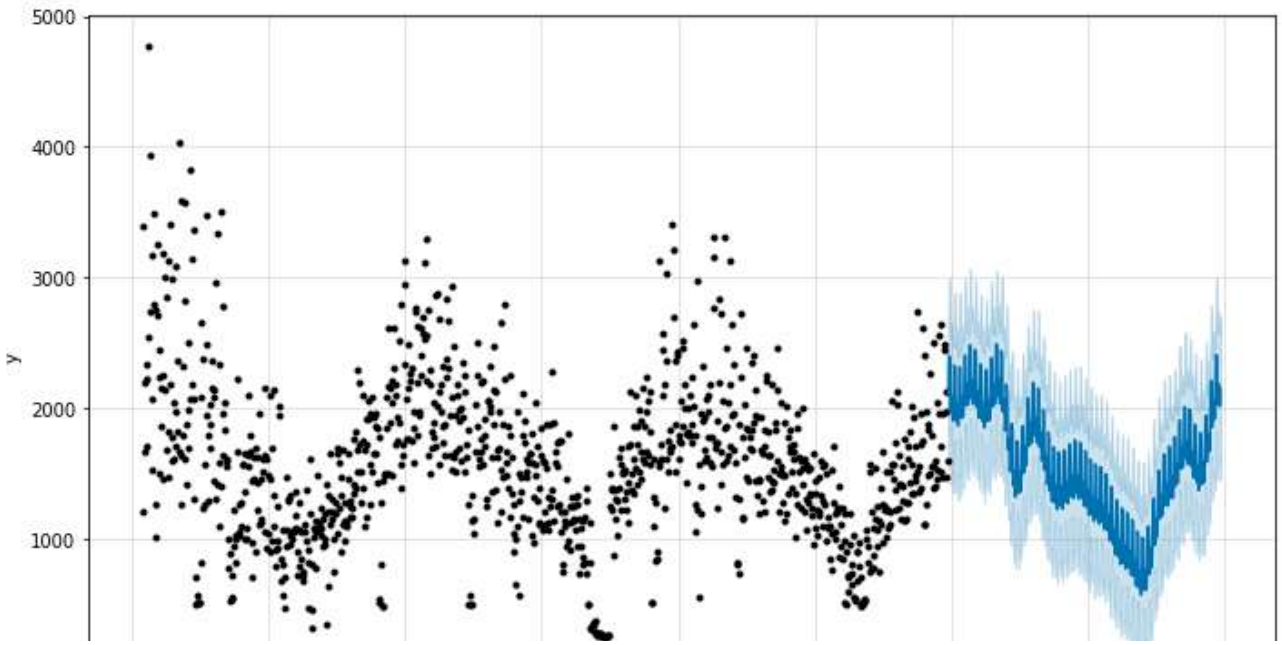
```
from fbprophet import Prophet
model=Prophet()
model.fit(X_train)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True
<fbprophet.forecaster.Prophet at 0x7f677077dc50>
```

```
forecast = model.predict(X_test)
forecast.head()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_te
0	2009-11-27	1580.084009	1538.098939	2655.419310	1580.084009	1580.084009	519.056
1	2009-11-28	1580.207768	1874.097743	2996.493841	1580.207768	1580.207768	822.213
2	2009-11-29	1580.331526	1781.560735	2936.906976	1580.331526	1580.331526	796.157
3	2009-11-30	1580.455285	1420.280691	2546.490800	1580.455285	1580.455285	396.118
4	2009-12-01	1580.579043	1526.008489	2662.526742	1580.579043	1580.579043	516.895

```
model.plot(forecast,uncertainty=True);
```



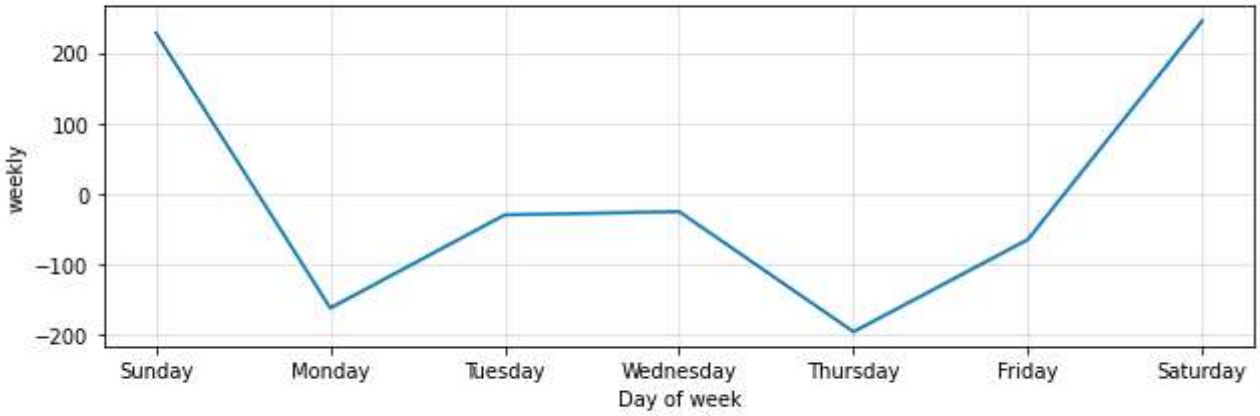

```
from sklearn import metrics
def mean_absolute_percentage_error(y,y_pred):
  y,y_pred = np.array(y), np.array(y_pred)
  return np.mean(np.abs((y - y_pred)/y)) *100
mape=mean_absolute_percentage_error(X_test['y'],forecast['yhat'])
mape
```

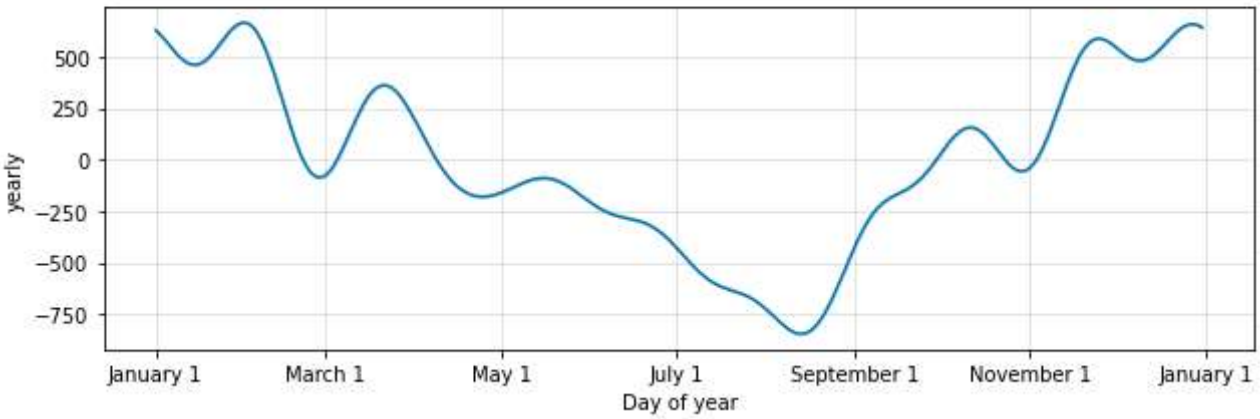
```
20.815831963532684
```

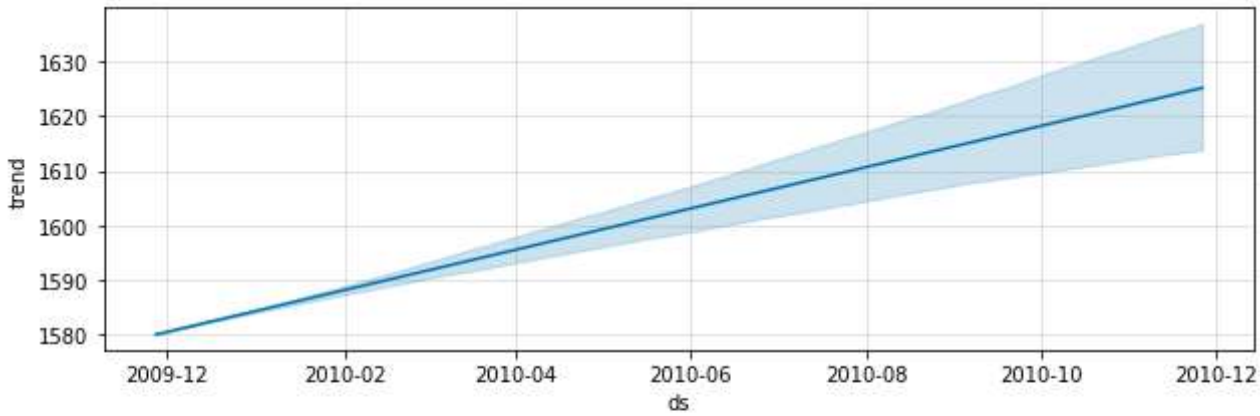
```
rmse = np.sqrt(metrics.mean_squared_error(X_test['y'],forecast['yhat']))
rmse
```

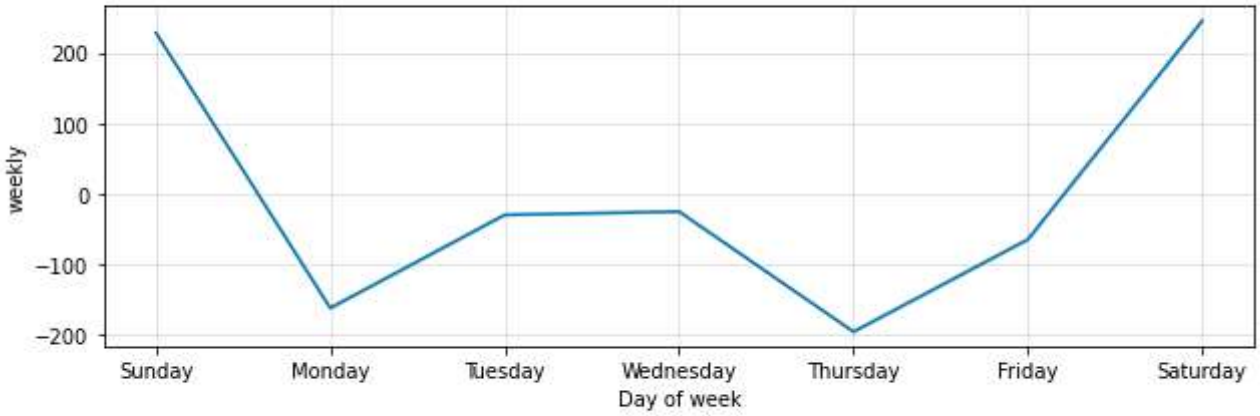
```
374.604233946713
```

```
model.plot_components(forecast)
```










```
resampled_data.head()
```

	datetime	Global_active_power	Global_reactive_power	Voltage	Global_intens
0	2006-12-16	1209.176025	34.922001	93552.53125	5180.799
1	2006-12-17	3390.459961	226.005997	345725.31250	14398.599

```
columns={'datetime':'ds','Global_active_power':'y',
        'Global_reactive_power':'add1','Voltage':'add2',
        'Global_intensity':'add3','Sub_metering_1':'add4',
        'Sub_metering_2':'add5','Sub_metering_3':'add6'};
new_data = resampled_data.rename(columns = columns)
new_data.head()
```

	ds	y	add1	add2	add3	add4	add5	add6
0	2006-12-16	1209.176025	34.922001	93552.53125	5180.799805	0.0	546.0	4926.0
1	2006-12-17	3390.459961	226.005997	345725.31250	14398.599609	2033.0	4187.0	13341.0
2	2006-12-18	2203.825928	161.792007	347373.62500	9247.200195	1063.0	2621.0	14018.0
3	2006-							

```
new_train=new_data[:-365]
new_test=new_data[-365:]


model = Prophet()
model.add_regressor('add1')
model.add_regressor('add2')
model.add_regressor('add3')
model.add_regressor('add4')
model.add_regressor('add5')
model.add_regressor('add6')
```

```
<fbprophet.forecaster.Prophet at 0x7f677a85ead0>
```

```
model = model.fit(new_train)
new_pred = model.predict(new_test)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True
```

```
new_pred.head()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	add1
0	2009-11-27	1597.207251	1362.956407	1397.388265	1597.207251	1597.207251	7.516635
1	2009-11-28	1597.347956	1845.279372	1877.083007	1597.347956	1597.347956	-8.857395
2	2009-11-29	1597.488661	1629.121530	1659.855539	1597.488661	1597.488661	-2.788241

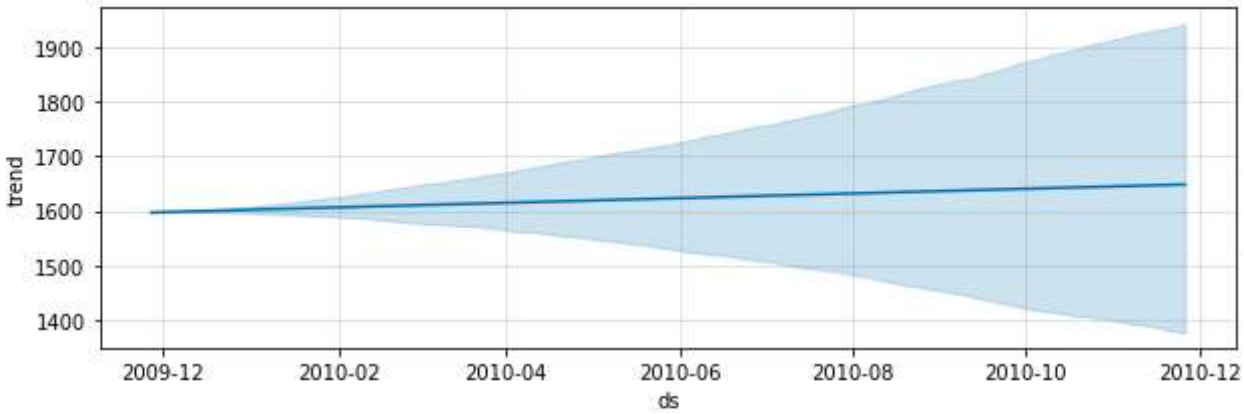
```
mape=mean_absolute_percentage_error(new_test['y'],new_pred['yhat'])
mape
```

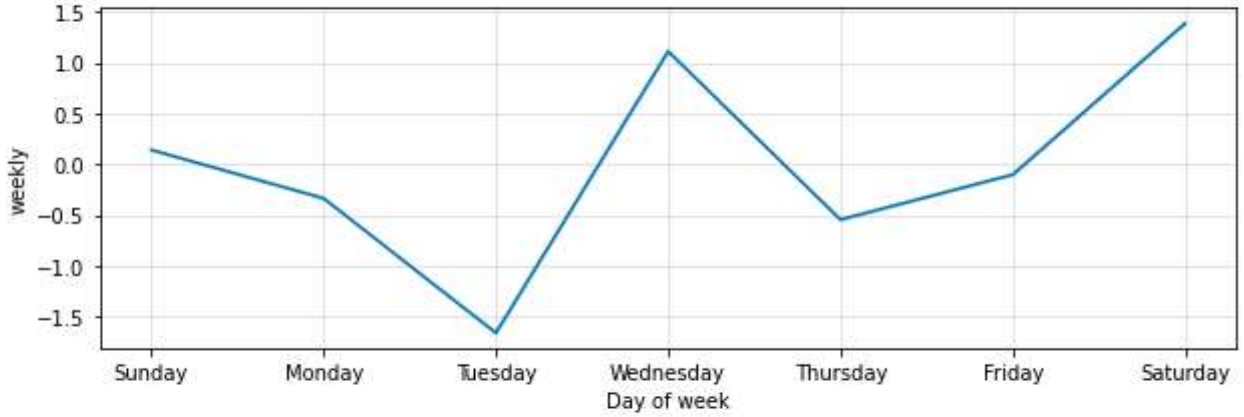
```
3.0445761457342444
```

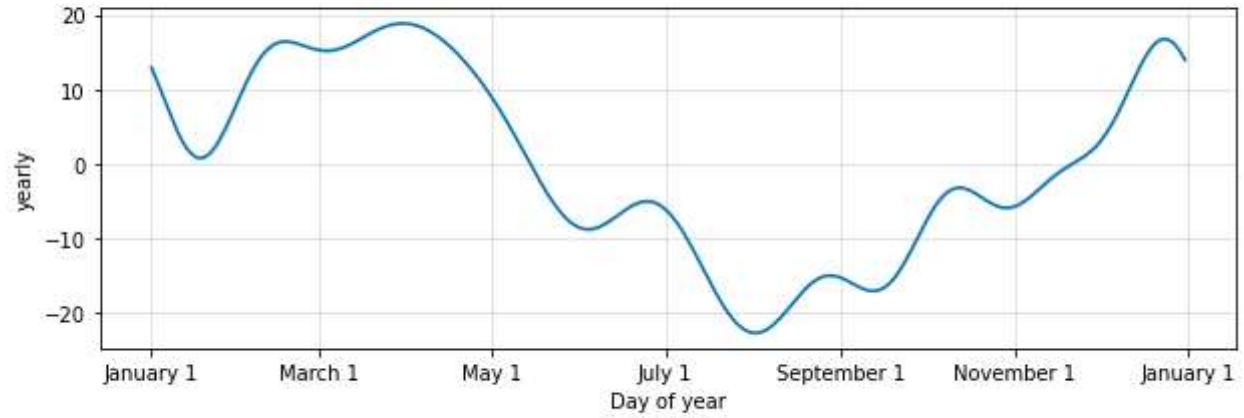
```
rmse = np.sqrt(metrics.mean_squared_error(new_test['y'],new_pred['yhat']))
rmse
```

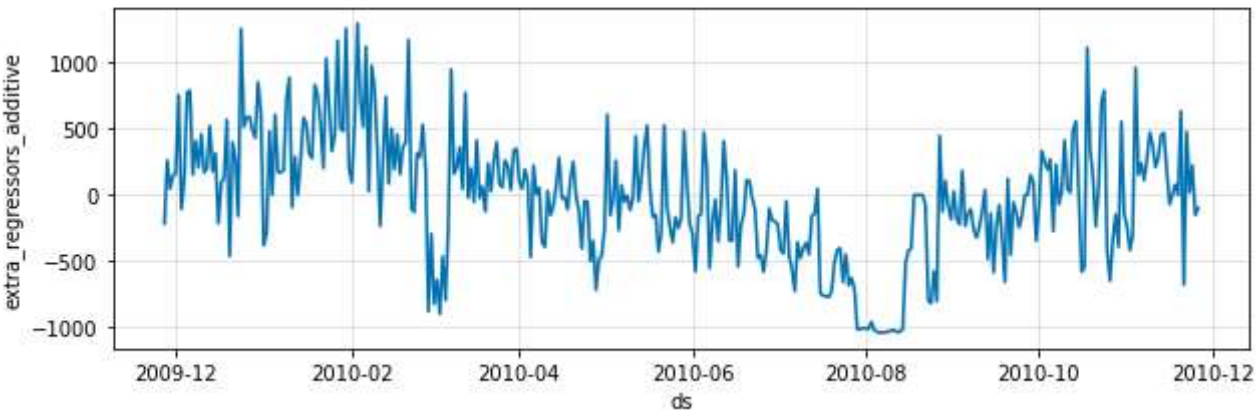
```
44.87909204303396
```

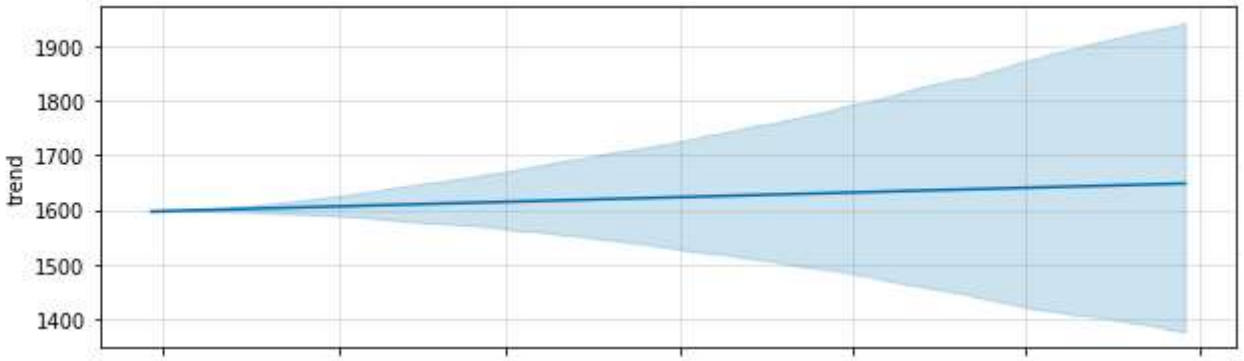
```
model.plot_components(new_pred)
```


trend
1900
1800
1700
1600
1500
1400
2009-12
2010-02
2010-04
2010-06
2010-08
2010-10
2010-12
ds


weekly
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week


yearly
20
10
0
−10
−20
January 1
March 1
May 1
July 1
September 1
November 1
January 1
Day of year


extra_regressors_additive
1000
500
0
−500
−1000
2009-12
2010-02
2010-04
2010-06
2010-08
2010-10
2010-12
ds


trend
1900
1800
1700
1600
1500
1400