```
from fbprophet import Prophet
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
import numpy as np
import io
from google.colab import files

data_to_load = files.upload()
df = pd.read_csv(io.BytesIO(data_to_load['AEP_hourly.csv']))
df['Datetime'] = pd.to_datetime(df['Datetime'], errors='coerce')
data = pd.DataFrame({'ds': df['Datetime'], 'y': df['AEP_MW']})
```

Choose Files AEP_hourly.csv

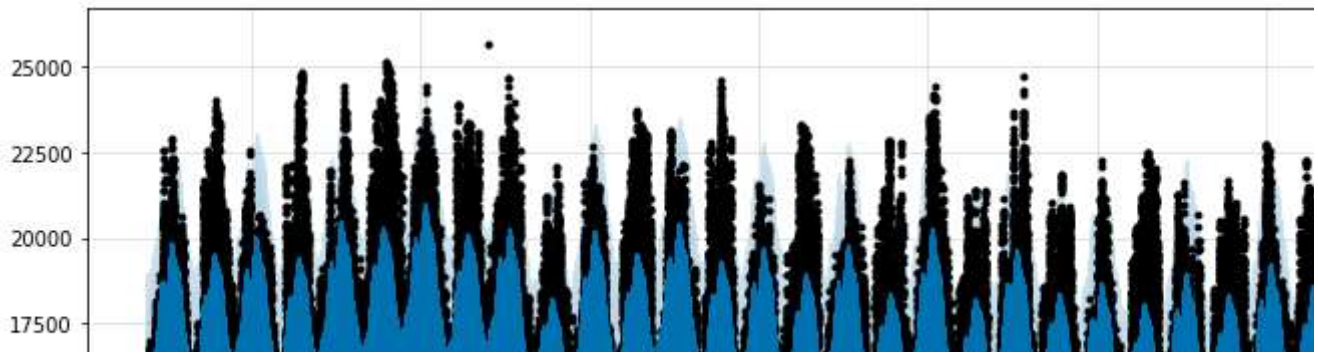
- **AEP_hourly.csv**(application/vnd.ms-excel) - 3516783 bytes, last modified: 11/29/2021 - 100% done

```
Saving AEP_hourly.csv to AEP_hourly.csv
```

```
from time import time
from sklearn.metrics import mean_squared_error


start = time()
m = Prophet(interval_width=0.95)
m.fit(data)
future = m.make_future_dataframe(periods=120)
forcast = m.predict(future)
end = time()
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
```

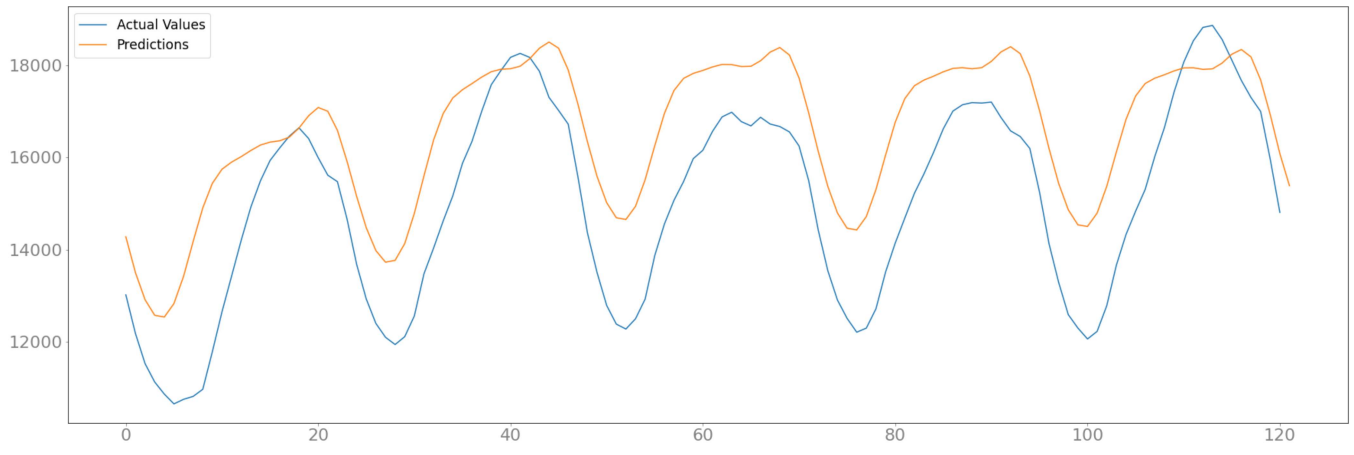
```
from math import sqrt
rmse = sqrt(mean_squared_error(data.loc[:, "y"], forcast.loc[:len(data["y"])-1, "yhat"]))


cfp = m.plot(forcast)
```




```
print("Test RMSE: %.3f" % rmse)
print("Total time taken to forecast: %.3f" % abs(start-end))
```

```
Test RMSE: 1642.646
Total time taken to forecast: 366.170
```


```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(30,10))
ax = fig.add_subplot(111)
ax.plot(data.loc[len(data["y"])-121:, "y"].values, label='Actual Values')
ax.plot(forcast.loc[len(forcast["yhat"])-151:len(forcast["yhat"])-30, "yhat"].values, label='
ax.tick_params(axis='x', colors='grey', labelsize=22)
ax.tick_params(axis='y', colors='grey', labelsize=22)
ax.legend(fontsize='xx-large')
plt.show()
```



✓ 0s completed at 1:38 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.