SMARTBRIDGE INTERNSHIP PROGRAM

Final Report

Team Epoch

SMARTBRIDGE

INTERNSHIP

PROGRAM

Web App for Forecasting Energy Output of Wind farms based on the Weather Conditions Using Prophet Algorithm with a Dashboard

Team Members

1. Shashwat Pandey
2. Dharana
3. Kshitij Singh

Table of Contents:

A. Main Model Code

1.Introduction:

1.1 Overview & Purpose :

Every country is aiming at economic stability and industrial growth. But while considering various avenues of interest to invest, the conventional power companies supersede the renewable energy sector in terms of Profit Margin.

This idea has huge potential to revolutionize the way people do their business. One of the major factors is that by implementing this model we can save a lot of **ancillary costs**.

- According to the length of the prediction horizon, wind speed forecasting can be classified into long-term forecasting and short-term forecasting. The former can provide critical information for site location, windmill planning, and proper wind turbine sections for specific wind farms .
- Precise short-term wind speed forecasts can minimize scheduling errors that can exert a large impact on grid reliability and market-based ancillary costs Because wind power is proportional to the cube of wind speeds and a 10% deviation of the expected wind speed results in an approximately 30% deviation in the expected wind power production , the prediction error of wind energy largely depends on the accuracy of wind speed forecasts.
- Accurate wind speed predictions for each of the farm's turbines are critical for the management of wind farms, which is usually the basis of wind power prediction and effective wind power utilization and can increase the reliability of the power grid and reduce operating costs.

Hence in order to have an efficient, working and profitable wind farm this web app is best suited in terms of ideal expenses on operational cost.

2. Literature Survey:

2.1 Existing Problems:

Here, reliability and failure prediction models can enable operators to apply preventive O&M strategies rather than corrective actions.

In order to develop these models, the failure rates and downtime of wind turbine (WT) components have to be understood profoundly. This project is focused on tackling three of the main issues related to WT failure analyses.

These are, the non-uniform data treatment, the scarcity of available failure analyses, and the lack of investigation on alternative data sources.

2.2 Proposed Solution:

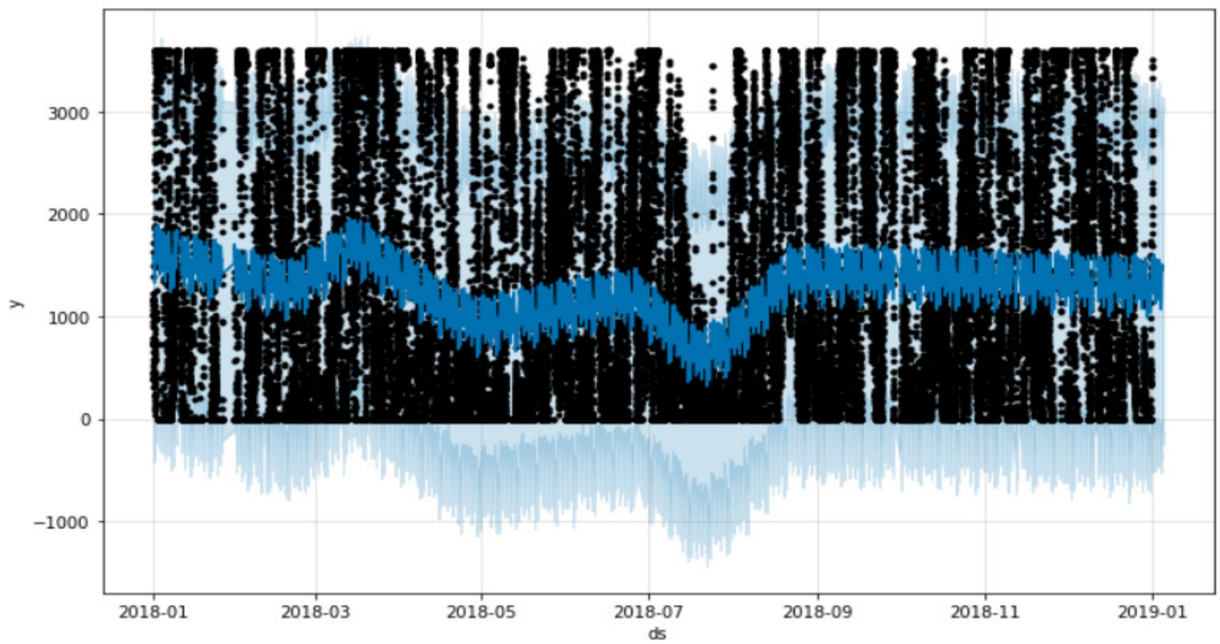
We tend to propose an idea of a Time Series Forecasting based Recommender web application that predicts the power requirement backed by Facebook's **Prophet**, by using Real -Time weather data.

It is a means for forecasting time series data based on trends fit with yearly, weekly, and daily seasonality, plus holiday effects. It is robust to missing data and shifts in the trend, and typically handles outliers well so that the usage of Wind energy is better analyzed.

DateTime	Pressure | (atm)	Wind speed | (m/s)	Wind direction | (deg)	Power generated by system | (kW)
2007-01-01 00:00:00	0.979103	9.014	229	33688.1
2007-01-01 01:00:00	0.979566	9.428	232	37261.9
2007-01-01 02:00:00	0.979937	8.700	236	30502.9
2007-01-01 03:00:00	0.980053	8.481	247	28419.2
2007-01-01 04:00:00	0.979867	8.383	256	27370.3

Visualization

The model architecture has been fully deployed as a web application using web framework Flask in Python, to forecast (1 hour -72 hour) power output, taken as input by the user in the web app.



Final stage

Recommendation app

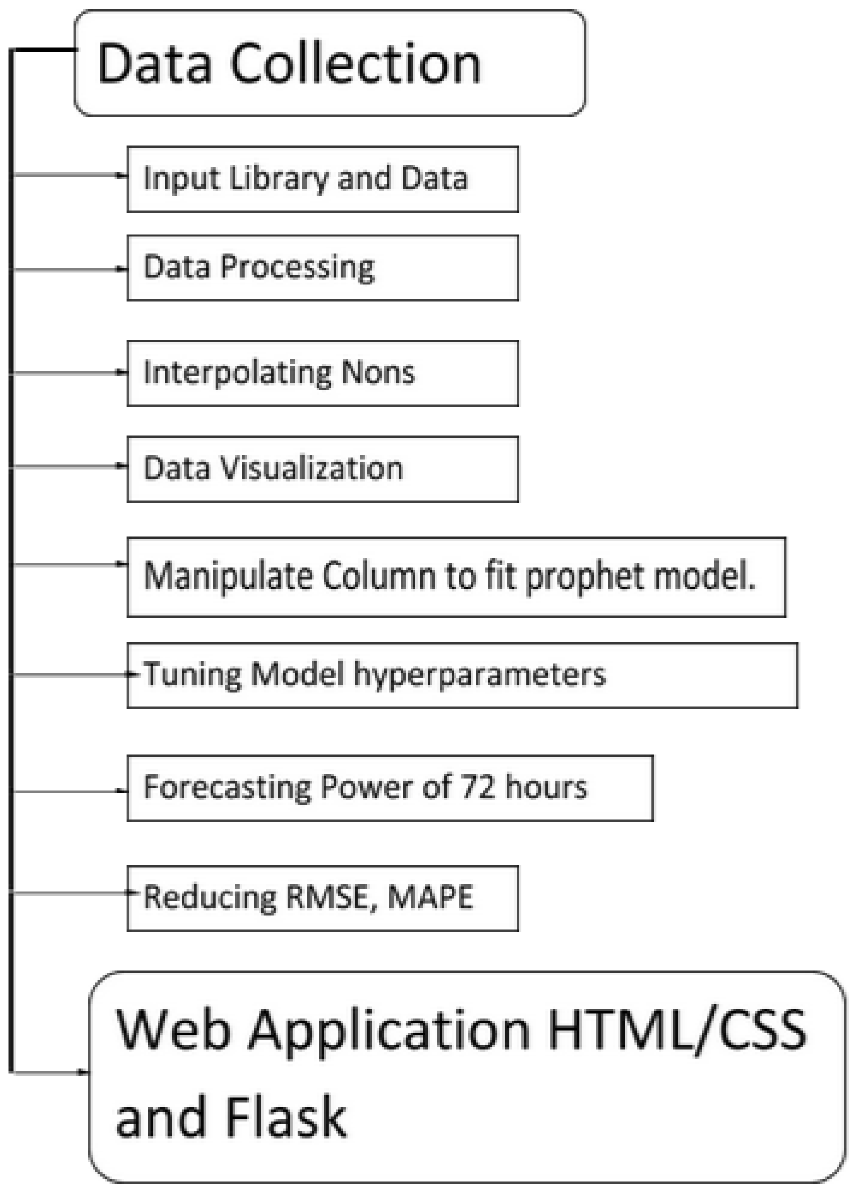
A python script will find the **Predicted Power Output using the model's pickle file** by using the weather data taken as input by the HTML page and determine which time interval is best suited for utilizing the production from wind farms to the Power Grid. This info is sent back to the HTML page and displayed to the user.

Dashboard

We also have a Node-RED UI for real time anemometer readings of Wind speeds and Wind Directions of 4 metropolitan cities.

3.Theoretical Analysis:

3.1 Block Diagram:



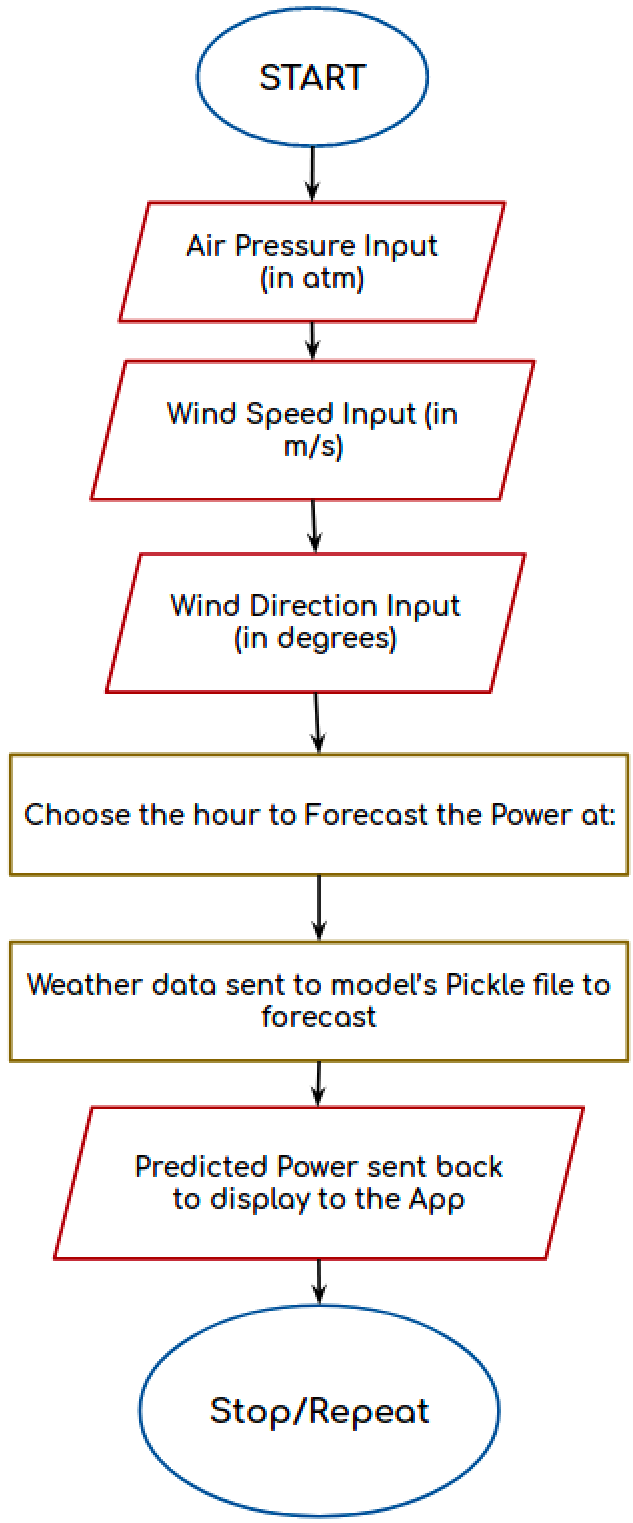
3.2 Hardware/Software Designing:

- Facebook's Prophet is an open source library
- Importing Libraries like Numpy, pandas and fbprophet
- Data Visualization using Matplotlib and seaborn in Model
- Flask based Web application for **CRUD API**
- Pygal library(in Python) for plotting predictions in web app
- HTML,CSS and JS for templates of **web app** and the **DASHBOARD**

4. Experimental Investigation:

We tuned a Prophet model's hyperparameters to reduce the RMSE, MAPE and make a time series forecasting model.
We used the value in columns **"Wind Direction, Wind Speed, Air Pressure" as features and "Active Power" as labels** for the model. Preprocessed the NaNs and interpolated the broken data. It was a 5 year weather data used to forecast power upto 72 hours.
The data used for making the forecasting model named as **'India.csv'** is taken from NREL database of India.

5. FlowChart:


START
Air Pressure Input
(in atm)
Wind Speed Input (in
m/s)
Wind Direction Input
(in degrees)
Choose the hour to Forecast the Power at:
Weather data sent to model's Pickle file to
forecast
Predicted Power sent back
to display to the App
Stop/Repeat

6. Result:

We built a Time Series Forecasting model from tuning the parameters **Prophet algorithm** to predict the Energy output of wind turbine using python. We made a Web Application that inputs Weather data and hour and predicts the value of power using Pickle file of the model.
We also have a **Node-RED UI** for real time anemometer readings of Wind speeds and Wind Directions of 4 metropolitan cities.

7.Advantages & Disadvantages:

7.1 Advantages:

- Wind power has a remarkably small impact upon the carbon footprint.
- There are zero carbon emissions associated with the operation of wind turbines.
- Accurate wind speed predictions for each of the farm's turbines are critical for the management of wind farms.
- Wind energy has one of the lowest water consumption footprints, unlike fossil fuels and nuclear power plants.
- Indeed it can exert a large impact on grid reliability and market-based ancillary costs.
- Hence in order to have an efficient, working and profitable wind farm this web app is best suited in terms of ideal expenses on operational cost.

7.2 Disadvantages:

- Permits are very difficult to get hold of for onshore wind farms due to the visual impact of the turbines.
- If the wind speed is less than threshold value, turbines depend on other forms of electricity generation in order to operate.

- Very high cost and complexity of manufacturing offshore wind farms makes the requirement of technical expertise necessary.
- Multiple wind turbines to be built in order to make an impact because Wind turbines generate a lot less power than the average fossil fuelled power station.

8.Applications:

- Web apps can be accessed by anyone having an internet connection and a device that supports browsers like smartphone, computer etc.
- No liberation of greenhouse gases makes wind energy environment friendly.
- More and more data will lead to better trend projections and accurate forecasts.
- Precise short-term wind speed forecasts can minimize scheduling errors.
- Effective wind power utilization can increase the reliability of the power grid and reduce operating costs.
- Variation in sizes and capacity of Wind Farms can be planned based on the needs of various populations.
- Forecasting the power beforehand gives an edge to the manufacturers to predict the approximate time period for Return on Investment of the Wind Farms.
- With the development of such apps like ours, efficiency of exploitation of wind energy has increased to a considerable amount.
- This web app can be used to find the right time to invest on Wind Farms based on the predictions

9.Conclusion:

The web app and the **Dashboard** can be used to Visualize various hourly prediction of Power from Wind farms. This data can be used by the Power Grid to judge when to use the wind farms and when not to. This will save a lot of energy and capital. Since it is deployed on the web it can be accessed by anyone, anytime and anywhere. It just requires internet connection.

10. Future Scope

1. The web app can be used to find out:
a. Trend Projection: This method used the underlying long-term trend of time series of data to forecast its future values.
b. Trend and Seasonal Components Method: This method uses a seasonal component of a time series in addition to the trend component.

2. The analysis of the Power output made by the app can be used to make build changes in the Wind Turbines to improve efficiency for the future.

3. More Data will lead to much accurate predictions and localised prediction based on the weather data of a particular area.

11. Bibliography

- **Dataset** - https://developer.nrel.gov/docs/wind/wind-toolkit/
- **Model and its implementation** - https://facebook.github.io/prophet/docs/quick_start.html
- **Templates -** https://developer.mozilla.org/en-US/
- **Flask applicaton -** https://flask.palletsprojects.com/en/1.1.x/
- **Node-RED -** https://nodered.org/

APPENDIX

Task 5345

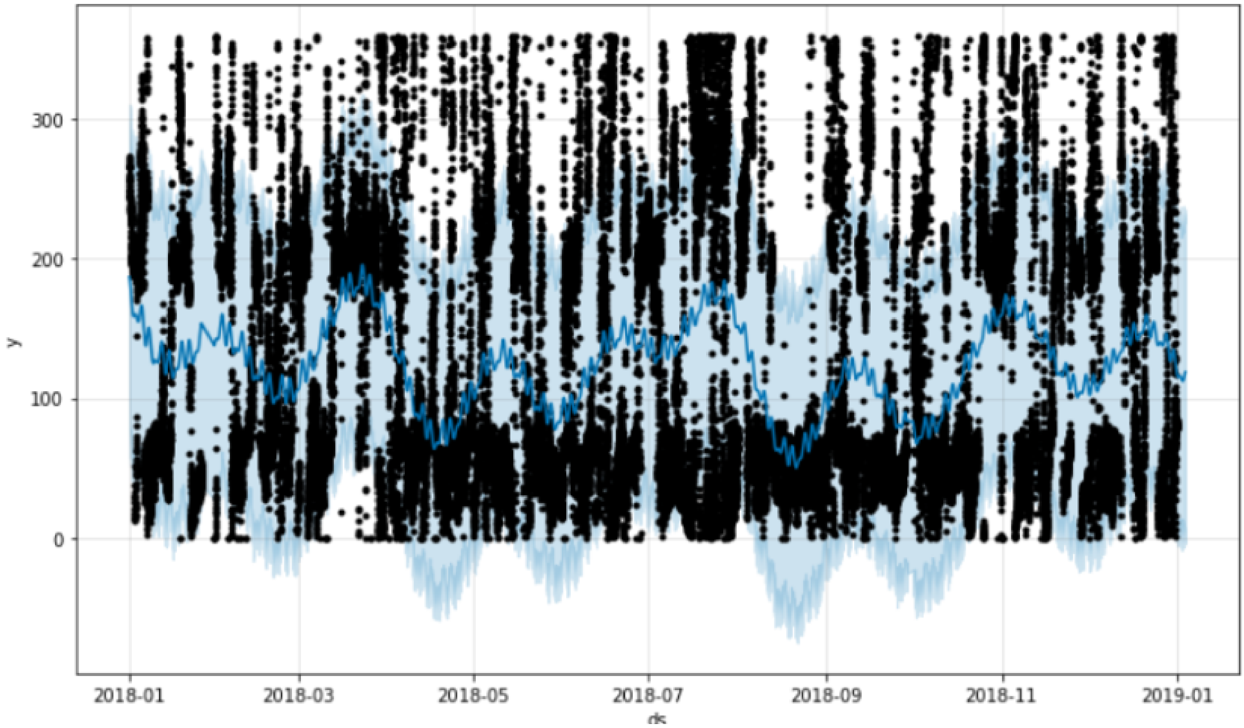
Preprocessing and Forecasting by Prophet Model

prophet.py

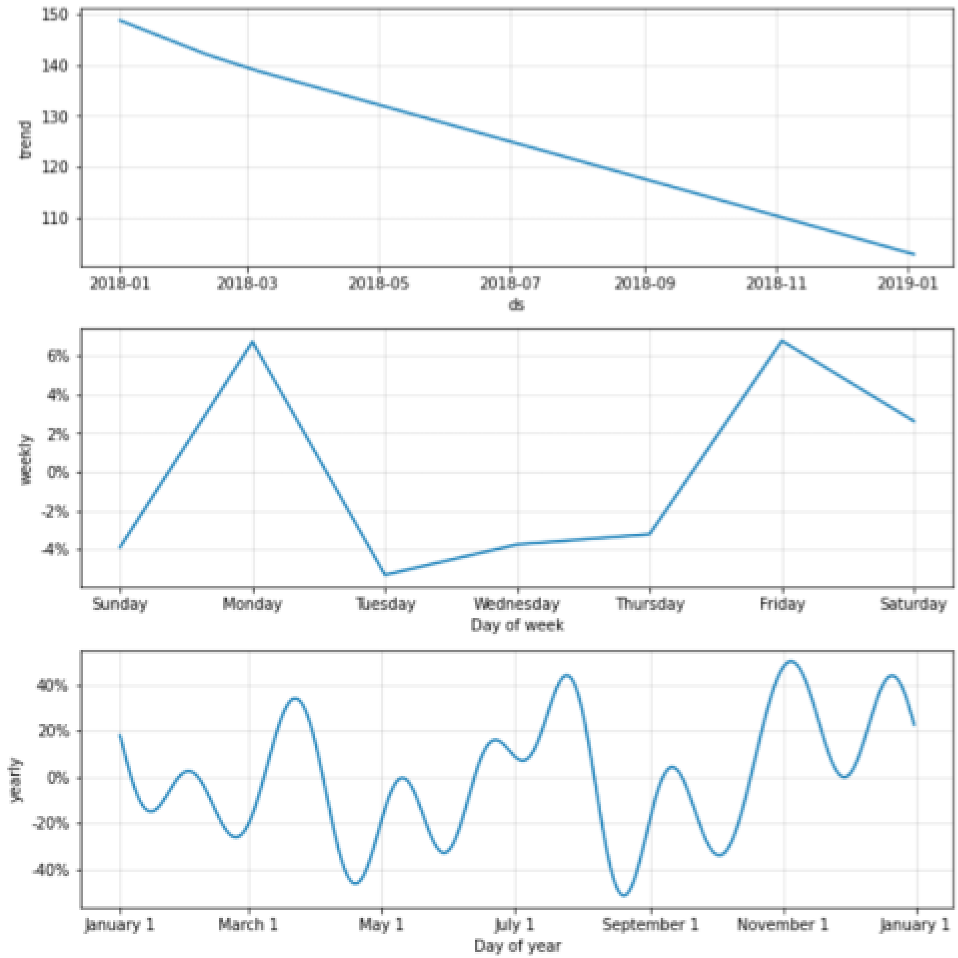
```
#!pip install -U fbprophet==0.5
# import pickle
# pkl_path = "/content/prophet_new.pkl"
# with open(pkl_path, "wb") as f:
#     pickle.dump(model, f)
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from fbprophet import Prophet
from fbprophet.plot import plot_plotly
from fbprophet.diagnostics import cross_validation
from fbprophet.diagnostics import performance_metrics
from fbprophet.plot import plot_cross_validation_metric

data = pd.read_csv('India.csv', infer_datetime_format=True)

data.head()

data.drop('Unnamed: 0', inplace=True, axis=1)

data.head()

data.replace(0, np.nan, inplace=True)

```

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

data.interpolate(method='linear', inplace=True)
data.head()

data.isnull().sum()

data.drop("Air temperature | ('C)", inplace=True, axis=1)
data.rename(columns={'DateTime':'ds', 'Power generated by system | (kW)':'y'}, inplace=True)
data.head()

data = data[["ds", "y", "Pressure | (atm)", "Wind speed | (m/s)", "Wind direction | (deg)"]]
data.head()

model = Prophet(mcmc_samples=0, seasonality_prior_scale=0.1,changepoint_prior_scale=0.05, seasonality_mode='multiplicative', \
            yearly_seasonality=20, \
            weekly_seasonality=False, \
            daily_seasonality=False)
model.add_regressor('Wind Speed (m/s)', prior_scale=0.5, mode='multiplicative', standardize=False)
model.add_regressor('Wind Direction (°)', prior_scale=0.5, mode='multiplicative', standardize=False)
model.fit(data)

fig = model.plot(forecast)
```

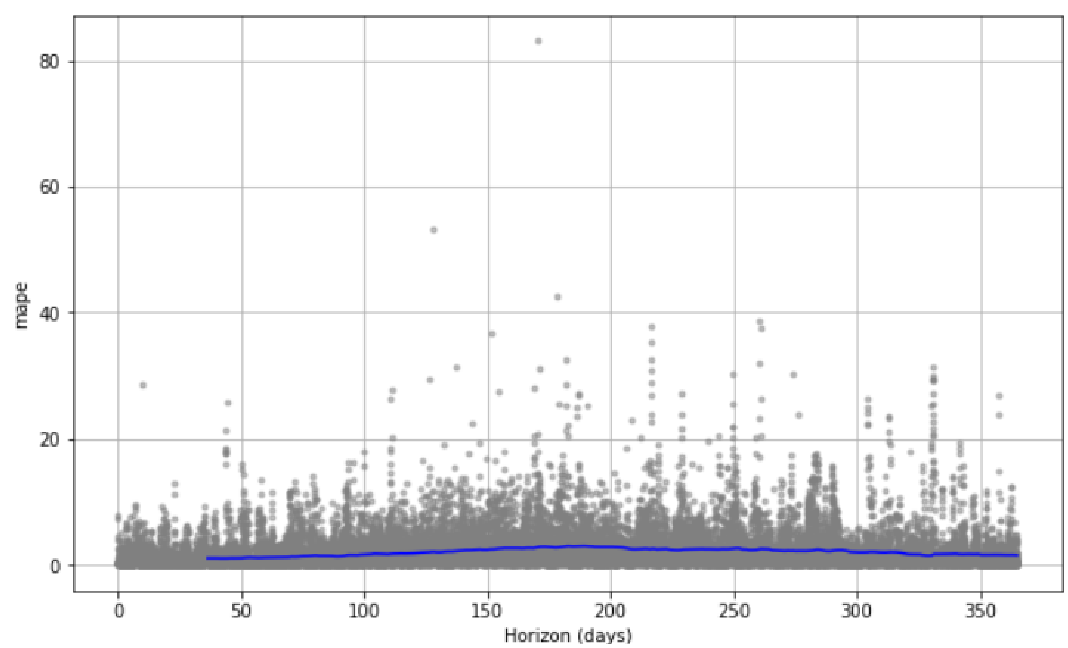
SMARTBRIDGE INTERNSHIP PROGRAM




```
fig2 = model.plot_components(forecast)
```




```
fig = plot_cross_validation_metric(df_cv, metric='mape')
```



Task 5346

Recommendation

Prediction by Backend

app.py

We are building the web application to input the data values of Wind Speed, Direction and Air Pressure--> Use the backend Flask app to access the Prophet Forecasting model ---> Predict the Next 1 hour Active Power prediction!

```
from flask import Flask,request,render_template, make_response, redirect
import pandas as pd
import pickle

model = pickle.load(open('prophet_new.pkl', 'rb'))

app = Flask(__name__)

@app.route('/')
def welcome():
    return render_template('welcome.html')

@app.route('/pred')
def pred():
    return render_template('prediction.html')

@app.route('/predict', methods =["POST","GET"])
def predict():
    # ===========-> Predicting power for values entered by the user <-===============
    date = request.form["date"]
    q = float(request.form["c"])
    p = float(request.form["a"])
```

```
    r = float(request.form["b"])
# Displaying Model's predictions!
    neq=pd.DataFrame()
    neq = pd.DataFrame([[date,q,p,r]],columns=['ds', 'Pressure | (atm)', 'Wind speed | (m/s)', 'Wind direction | (deg)'], dtype=float)
     # ===========-> Actual Forecasting <-===================
    l = model.predict(neq)
    dk=l['yhat']
    out = dk[0]
    return render_template("prediction.html",dks=out)
@app.route('/dash')
def dash():
    return redirect('https://prophecy.eu-gb.mybluemix.net/ui/#!/1?socketid=ESU0DYjSMIcQJG3GAAC-')

if __name__ == '__main__':
    app.run(debug=True)
```

Task 5342

Initialise Flask app locally

flask.py

```
from flask import Flask, render_template, request, jsonify
from datetime import datetime
from fbprophet import Prophet
import pygal
import pandas as pd
import numpy as np

app = Flask(__name__)
@app.route('/home')
def hello():
```

```
    return "Hello World!"
if __name__ == '__main__':
    app.run(debug=True)
```

Task 5346

Node-RED UI