WIND OUTPUT ENERGY PREDICTION

By Aleena Umar

Introduction

Wind energy plays an increasing role in the supply of energy world wide. The energy output of a wind farm is highly dependent on the wind conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction.

Thus we aim to build a model which can predict this accurately as compared to present solutions.

We will be developing a machine learning model to Predict the power output of wind farm based on the previous data.

This prediction and other useful features related to this will be incorporated into an application for user display.

Overview : Presenting a UI cabable of prediction wind output energy depending upon given input parameters. As well as displaying current weather parameters-temp,humidity,pressure etc of wind farm site.

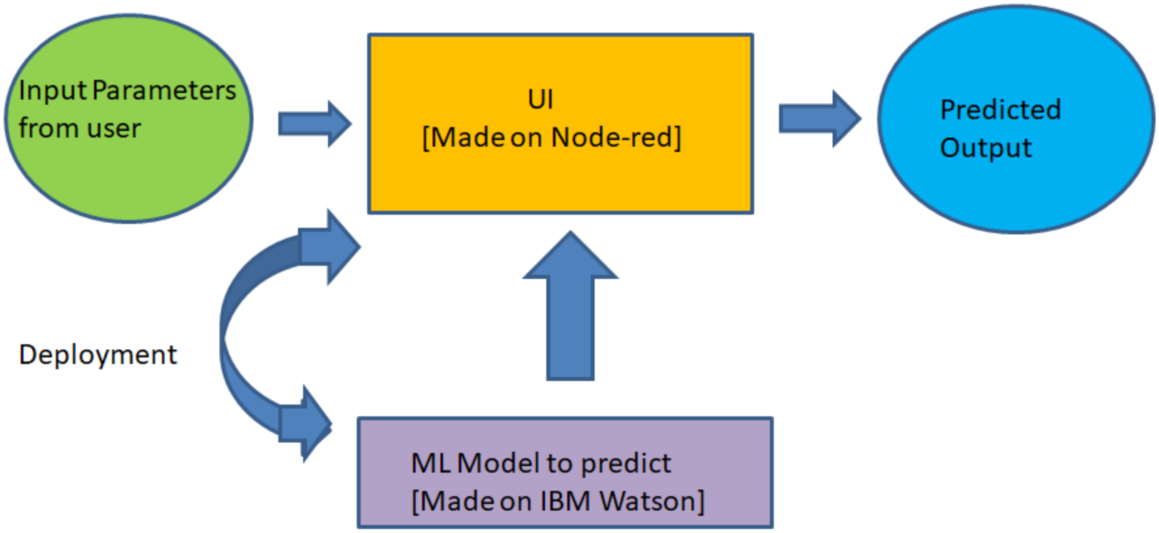
Purpose : The purpose is to optimise wind energy,which is a clean energy source.

Literature Survey :

Present problem : wind output power is difficult to predict due to unpredictable nature of the parameters such as wind speed. Wind energy plays an increasing role in the supply of energy world wide. The energy output of a wind farm is highly dependent on the wind conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction..

Proposed solution : By building a machine learning model, using previous data set for training we can predict the output wind energy, which can help in optimisation of the wind output energy.

FLOW CHART :



Advatages & Applications :

- As electricity and power is the backbone of any industry and if it is generated from a renewable resource, it makes the working all the more efficient. If energy output of wind farm can be predicted it can be optimised, which in long run leads to :-
- Growth of economy
- A great move towards sustainable development.
- Meeting the power requirements of all places.
- Enhance the present production.

Experimental Investigation

Data Collection : Load Data , Data Visualize , Feature Selection

LOAD DATA :-

For wind output energy prediction we collected data from Kaggle. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

https://www.kaggle.com/berkerisen/wind-turbine-scada-dataset

Import Data :

After downlaoding the Wind Dataset from above link we Import Data into our Jupityer Notebook /Watson Studio Notebook By Following Code

Jupityer Notebook

```
df = pd.read_csv('C:\Users\Aleena Umar\Downloads\T1.csv',encoding = "utf-8")
```

Watson Studio Notebook

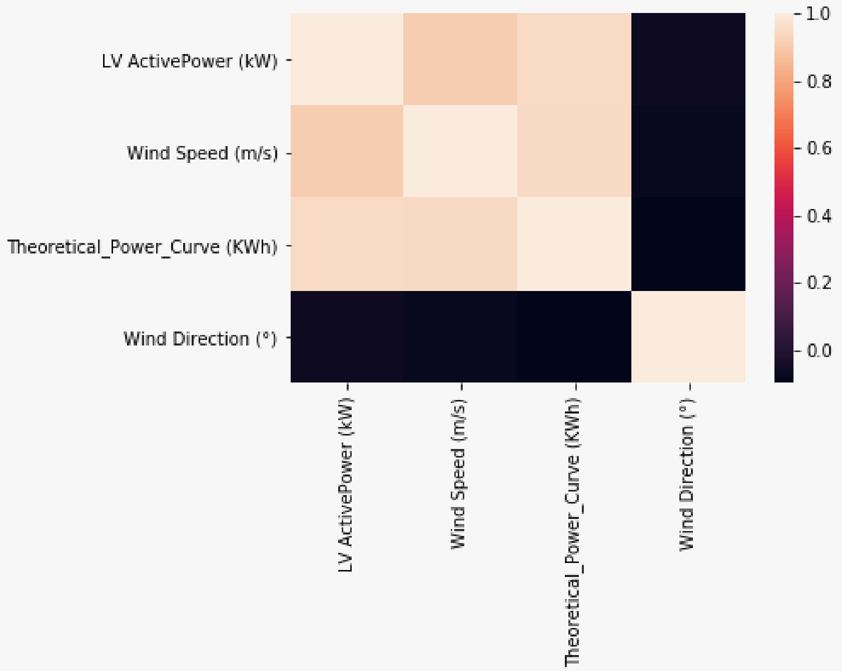
First I upload dataset on cloud stroage of IBM Cloud and from right section of notebook i import csv file as Pandas Dataframe and following code generated automatically in watson notebook.

```
importtypes
importpandasaspd
frombotocore.clientimport Config
importibm_boto3

def__iter__(self): return0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_9c69e53f6bf24045a870b6ce82b683ec = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='8TFJ2E4o2q72jqru2421UKRyvS_JNFqM0ICwVmI8C08G',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body =
client_9c69e53f6bf24045a870b6ce82b683ec.get_object(Bucket='newwind-donotdelete-pr-0dk3fsjy2
zdd3m',Key='T1.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
ifnothasattr(body, "__iter__"): body.__iter__= types.MethodType( __iter__, body )
df = pd.read_csv(body,encoding='unicode_escape',sep='\t)
```

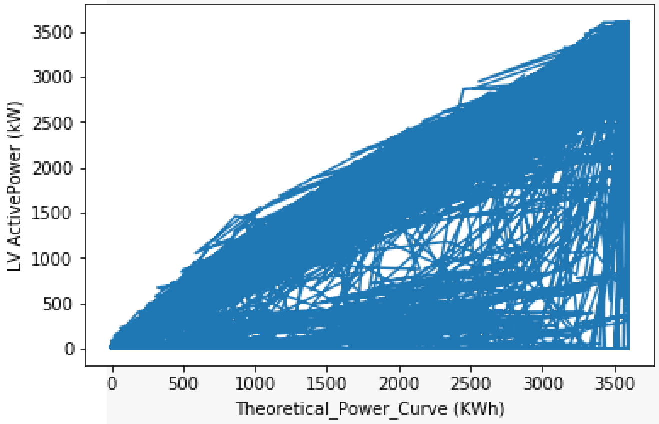
Data Visualize :

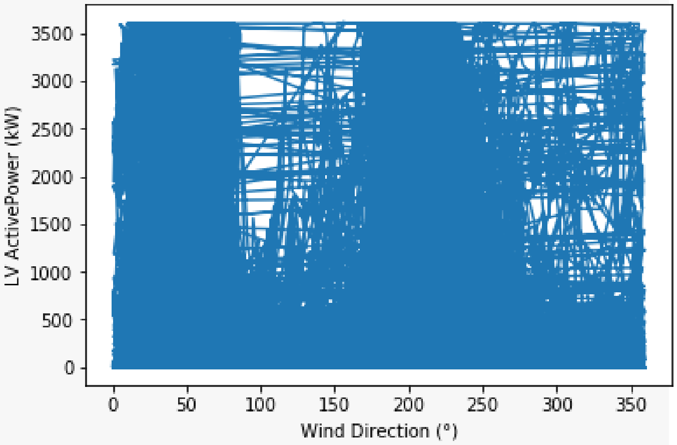
To Relate all Features of Data set and Find the what kind of correlation is held by attributes of data or find which features correspond to output we simply draw multivariate plots between all features and check from plots which feature is correspond to output.

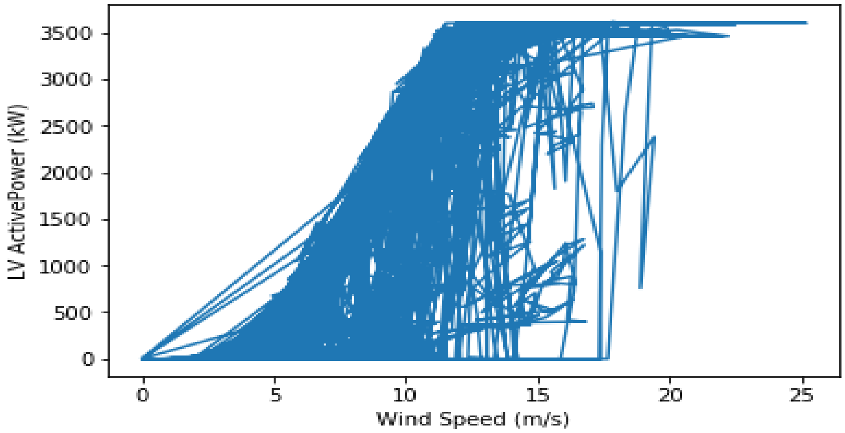
For correlate all features we use seaborn heatmaps and draw heatmap by using sns.heatmap(df.corr()) command.



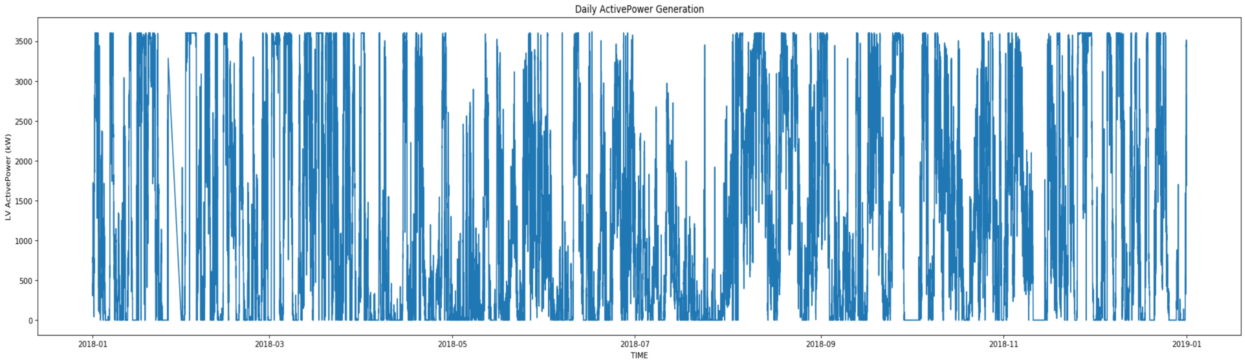
From above heatmaps we can view the correlation between all the features and we observe that Theoretical Power is very corresponding to ouput. Further we also draw individual plots of all features by matplotlib.



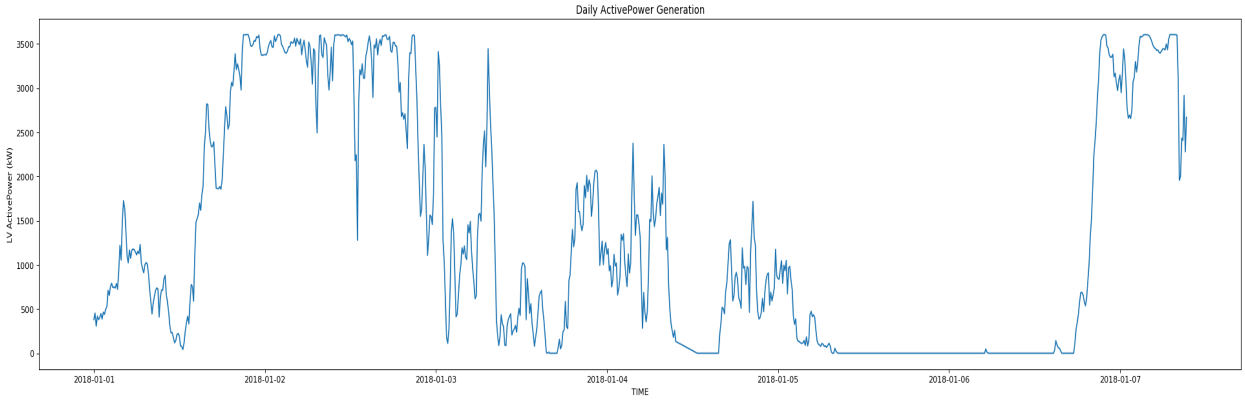




This is the graph we plot to observe the relation between Wind Speed and the output(LV Active Power). We also Developed Time Series Graph To See How our Prediction is vary according to Date and Time. Here Below Graph is Between show relation between time and LV active power (Yearly)



Here Below Graph is Between show relation between time and LV active power (Weekly)



Feature Selection :

Select Best Feature for model

5. **Building Pipeline :** We build the pipeline using linear regression, support vector machines and XGboost and check which algorithm gives better accuracy.

```
pipeline_lr=Pipeline([('scalar1',StandardScaler()),
        ('lr_classifier',LinearRegression())])
pipeline_svm=Pipeline([('scalar2',StandardScaler()),
        ('dt_classifier',svm.SVR())])
pipeline_xgb=Pipeline([('scalar3',StandardScaler()),
        ('rf_classifier',xg.XGBRegressor())])
```

6. Split Dataset into Train Data and Test Data :

```
x= df.iloc[:,2:5].values
y=df.iloc[:,1].values
```

7. Train Model

```
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.15, random_state=42)
```

```
my_model=xg.XGBRegressor()
my_model.fit(x,y)
```

8. Test the Model using Test dataset

```
y_pred = my_model.predict
```

Deploy Model :

1. Create Machine Learning Credentials

```
        from watson_machine_learning_clientimport WatsonMachineLearningAPIClient

wml_credientials = {
 "apikey": "zVaOw8hdN2LPBOonkfo87D_be0eswxeoneNu5Q1q1X0a",
  "iam_apikey_description": "Auto-generated for key 89050668-9c63-4316-be95-d2d04bc4ba4a",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn":
"crn:v1:bluemix:public:iam-identity::a/30db80e933a740d4907e2d1fe83e38ea::serviceid:ServiceId-fc
1192e0-70a7-4113-bde1-5abc761f17e4",
  "instance_id": "384fab1d-7994-43bb-9cfd-9b064756482b",
  "url": "https://eu-gb.ml.cloud.ibm.com"

}



        client = WatsonMachineLearningAPIClient(wml_credientials)
```

2. Create Metadata

```
metadata = {
                     client.repository.ModelMetaNames.NAME : 'Wind Predication',
                    client.repository.ModelMetaNames.AUTHOR_NAME : 'Aleena Umar',
                    client.repository.ModelMetaNames.AUTHOR_EMAIL :
'si05202001322@smartinternz.com
        '
        }
```

3. Save Model

```
stored_data = client.repository.store_model(my_model,meta_props = metadata)
```

4. Extract GUID

```
guid = client.repository.get_model_uid(stored_data)
```

5. Deploy Model from GUID

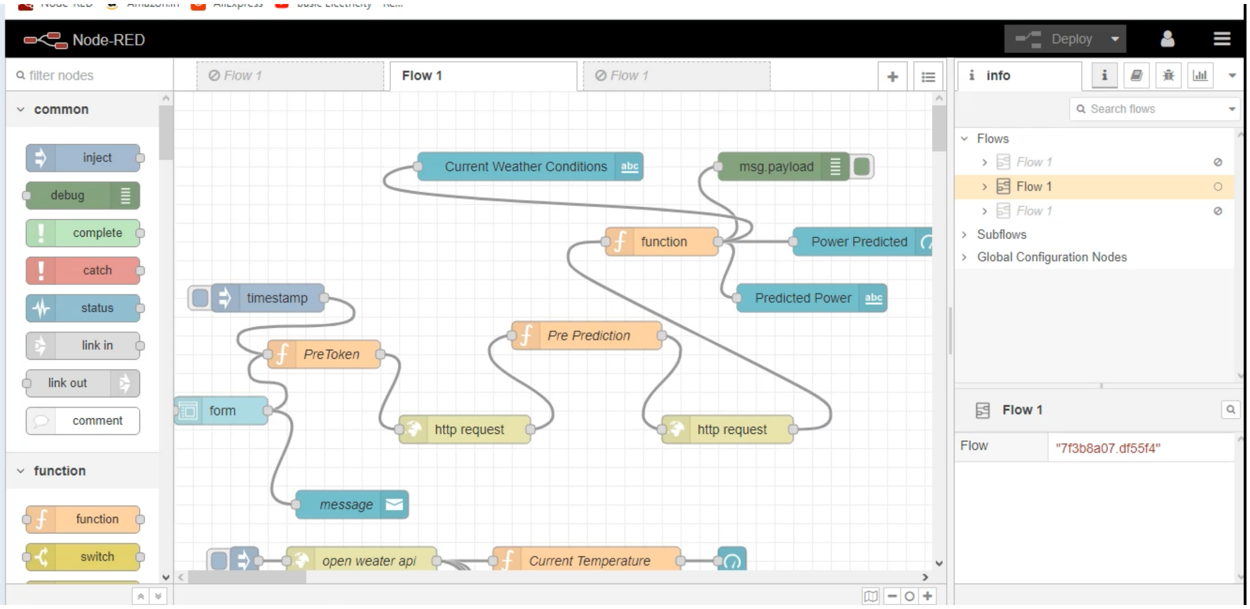
```
deploy = client.deployments.create(guid)
#########################################################################
#####################

     Synchronous deployment creation for uid: 'ff722eff-dfb1-4da6-b8da-ac10c64c5700
      '          started

     #########################################################################
#####################


      INITIALIZING
     DEPLOY_SUCCESS


     ------------------------------------------------------------------------------------------------
     Successfully finished deployment creation,
deployment_uid='dde44d34-99b0-4f8b-a9ee-cbcd8213ef14
      '
     ------------------------------------------------------------------------------------------------
```

Building UI Application :-

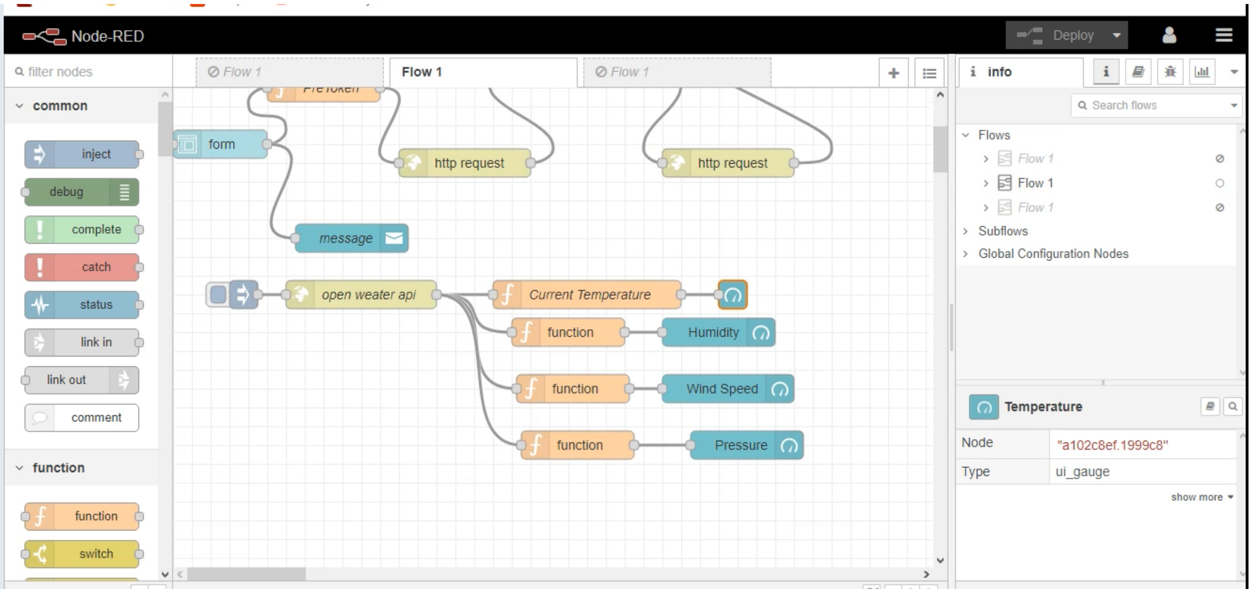
Accessing Node-red on IBM cloud :- select node-red service from catalogue and proceed by entering necessary information.

After the app is creadted visit on app url.

And create the node-red flow.

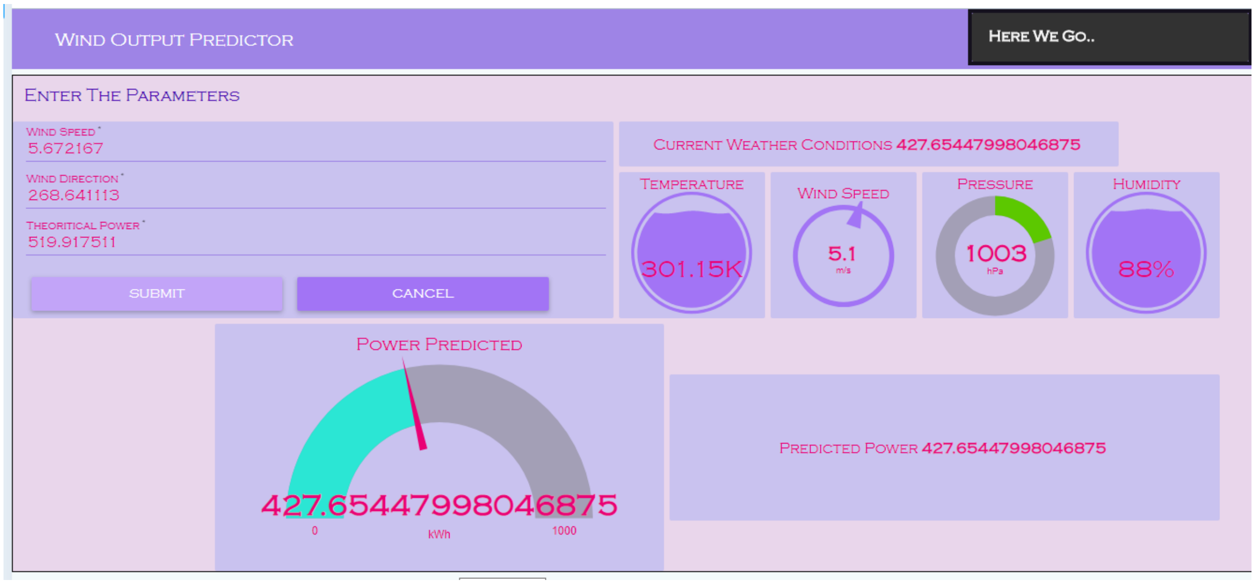


Flow "7f3b8a07.df55f4"



Node "a102c8ef.1999c8"

Type ui_gauge


Wind Output Predictor

Here We Go..

Enter The Parameters

Wind Speed *
5.672167

Wind Direction *
268.641113

Theoretical Power *
519.917511

Current Weather Conditions 427.65447998046875

Predicted Power 427.65447998046875

Deploying the node-red flow

Drag and edit the nodes from the node palettes as per the requirements of the UI like form node, message node,gauge node,text node etc.

FOR DEPLOYMENT

Drah the https request node and select moethod as POST and enter your scoring url from the notebook created at IBM watson in url blank.
Drag a function node and edit it's instance id as per it is given in the notebook and edit the variable as per declred as input in form node.
Drag another function node and write the below code:-

```
msg.payload=msg.payload.values[0][0]
return msg
```

Then drag the debug node
Connect all of these together and check the output.

RESULT : Successfully built the UI capable of predicting wind output energy.

Future Scope :

❄️If the output energy of the find farms will be predicted then we can open gateways to many brighter opportunities for our country like :-

❄️Increase employment
Move towards sustainable development.

BIBLIOGRAPHY & APPENDIX

My Github Link - IBM-Wind-Energy-Output-Prediction

THANKYOU

BY : ALEENA UMAR