

PROJECT REPORT

Predicting The Energy Output Of Wind Turbine

Based on Weather Condition

**Team Name: Data Pirates**

**Team Members:**

* Pradhuman Gupta (pradhumangupta099@gmail.com)
* Abhinav Jain (abhinavj65@gmail.com)
* Harshita Malhotra (harshita44441@gmail.com)
* Vinay Bansal (vina.bansal@gmail.com)

Category: Machine Learning

College: HMR Institute Of Techynology And Management

Webpage Link:

<https://node-red-qrqvq.eu-gb.mybluemix.net/ui/#!/6?socketid=v8uC89j4lN9nNvOgAAAX>

|  |
| --- |
| 1. INTRODUCTION |

* 1. **OVERVIEW**

A typical Regression Machine Learning project leverages data to predict insights into the future. This problem statement is aimed at predicting the wind speed given various features.

It makes use of Random Forrest for wind direction and Linear Regression for wind speed.

Focus of this research is on the analysis of behavior of the wind, with an aim to develop a wind prediction system with machine learning which can be used for production prediction of wind power plants. Wind is a free energy source; however it is highly unpredictable-which is a significant problem for integration of large wind power plants into an energy system. Wind data is a time series type of data. Therefore , two models will be considered: one for prediction of wind direction, and one for wind speed.

**Project Requirements**: Python, IBM Cloud, IBM Watson

**Functional Requirements**: IBM cloud

**Technical Requirements**: ML, WATSON Studio, Python, Node-Red

**Software Requirements**: Watson Studio, Node-Red

**Project Deliverables**: IBM Hack Challange

**Project Team**: Pradhuman Gupta, Abhinav Jain, Harshita Malhotra, Vinaay Bansal

**Project Duration**: 1 Month

* 1. **PURPOSE**

The result of this Wind Speed should not be interpreted as definitive. Actual longevity is based on many factors, not all of which are captured here. This will give information about enery produced at a particular time based on Wind Speed and Wind Direction.

|  |
| --- |
| 1. LITERATURE SURVEY |

* 1. **EXISTING PROBLEM**

A typical Regression Machine Learning project leverages data to predict insights into the future. This problem statement is aimed at predicting the wind speed given various features.

It makes use of Random Forrest for wind direction and Linear Regression for wind speed.

Focus of this research is on the analysis of behavior of the wind, with an aim to develop a wind prediction system with machine learning which can be used for production prediction of wind power plants. Wind is a free energy source; however it is highly unpredictable-which is a significant problem for integration of large wind power plants into an energy system. Wind data is a time series type of data. Therefore, two models will be considered: one for prediction of wind direction, and one for wind speed.

* 1. **PROPOSED SOLUTION**

**STEPS:**

a) Create IBM cloud services

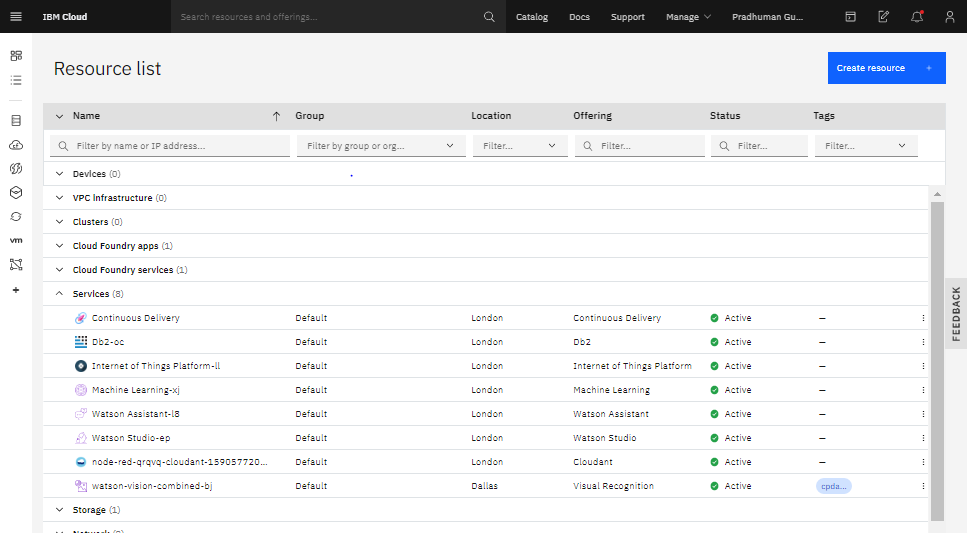
b) Configure Watson Studio

c) Create Node-Red Flow to connect all services together

d) Deploy and run Node-Red app

1. **Create IBM Cloud Services**

* Watson Studio
* Machine Learning
* Node-RED



1. **2.2.2) Configure Watson Studio**

After creating all services, Go to resource list and launch watson studio then get started with watson studio. Then create an empty project and add machine learning resource as associated services in settings. Create a token as editor type.

Then add dataset and empty jupyter notebook into Assets.

After that go to notebook and write your code to build model and get the scoring endpoint URL.

**STEPS FOR NOTEBOOK:**

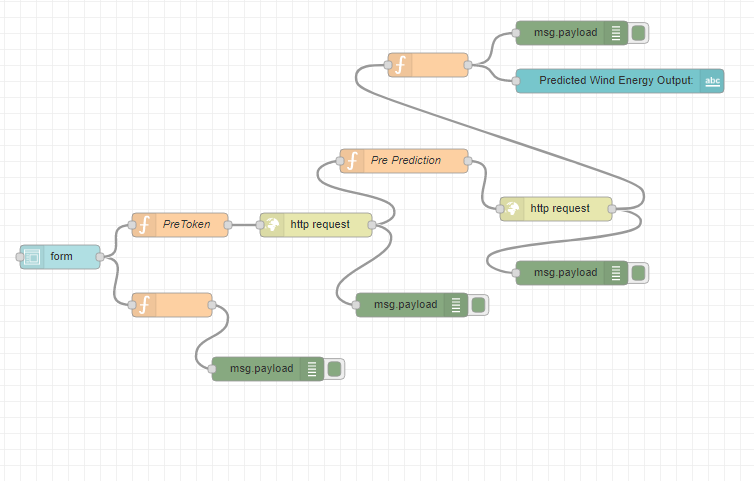
* Install Watson\_Machine\_Learning\_Client
* Import necessary Libraries
* Import DataSet.
* Data Processing
* Removing unusual species in column names using rename function.
* Replacing NAN values with their Mean values.
* Exploratory Data Analysis
* Plotting **Pair plot** for analysing pairwise relationship among features.
* Ploting a **HEATMAP** to check nif Dimensional Reduction can be Performed.
* Train And Test
* The dataset was splitted into two parts i.e Input and Output. As Wind Energy needs to be predicted so it is to be treated as output and all other columns are treated as Input
* Afterwards as we need regression technique to build our model so each and every column needs to be numeric . So then we check for numeric and categoric columns .
* Then train and test split was peformed and 80% of dataset were trained data and 20% were test data.
* Linear Regression
* In statistics, **linear regression** is a **linear** approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple **linear regression**.
* Model Building and Deployment
* At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient .

****

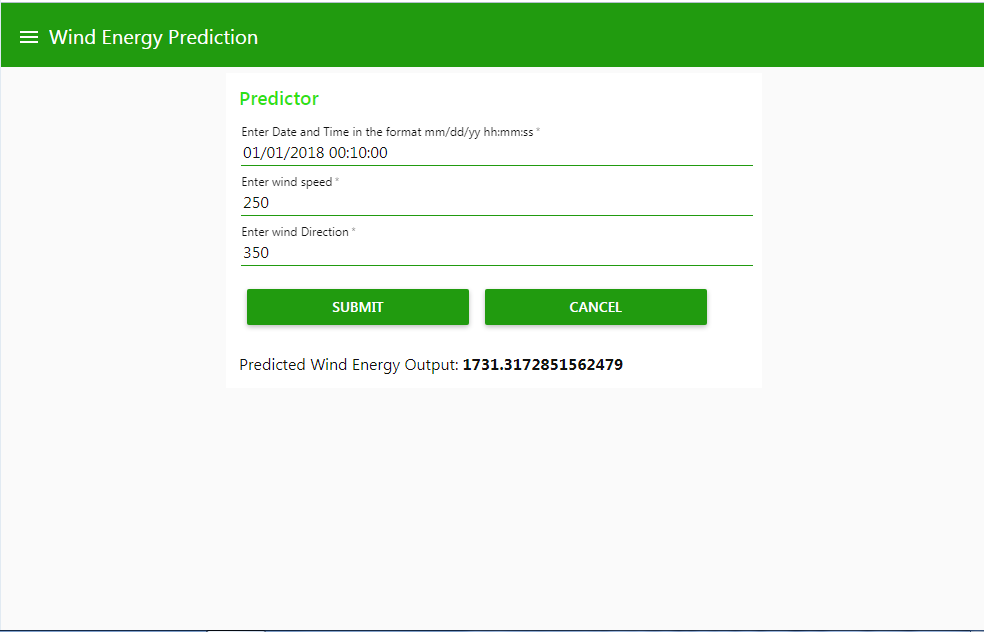
* Then the model was build and stored in model\_artifact.
* Then the model was deployed and scoring\_endpoint url was generated

1. **2.2.3) Create Node-RED Flow to Connect all Services together**

* Go to Node-RED Editor from Resource List.
* Install Node-RED Dashboard from Manage Pallete.
* Now Create the Flow With the Help of Following Flow:
* Inject
* UI\_Form
* Function
* Http\_Request
* Debug
* UI\_Text

****

* Deploy and Run Node-RED App.
* Deploy the Node Red flow. Then copy the link URL upto .net/ and paste at a new tab by UI at the end of the URL like this.

****

|  |
| --- |
| 1. THEORETICAL ANALYSIS |

**3.1) BLOCK DIAGRAM**

Input values to the fields such as ‘Enter Date and Time in the Format mm/dd/yy hh:mm:ss’, ‘Enter Wind Speed’, ‘Enter Wind Direction’ to the blank fields in webpage.

Predicting The Energy Output Of Wind Turbine Based on Weather Condition

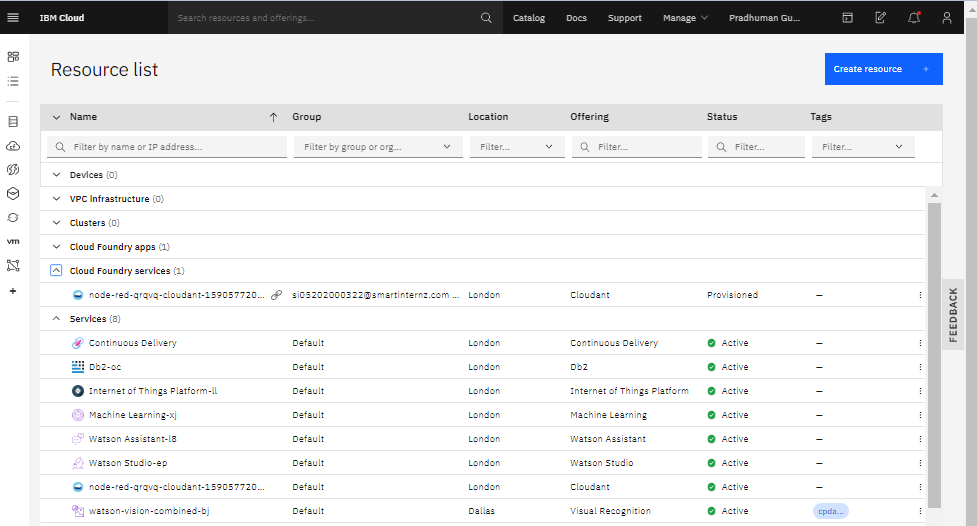
Deployed machine learning model with maximum accuracy score.

**3.2) HARDWARE/SOFTWARE DESIGNING**

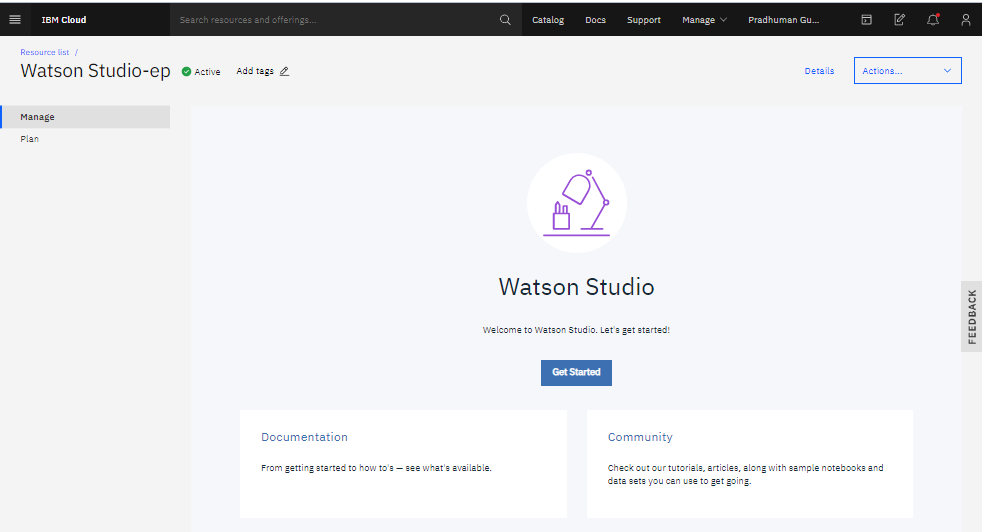
* **PROJECT REQUIREMENTS**
* Python
* IBM Cloud
* IBM Watson
* **FUNCTIONAL REQUIREMENTS**
* IBM Cloud
* **TECHNICAL REQUIREMENTS**
* Python
* IBM Watson
* IBM Cloud
* Machine Learning
* AutoAI
* **SOFTWARE REQUIREMENTS**
* IDLE (Python 3.8 )
* Jupyter Notebook
* IBM Cloud
* IBM Watson

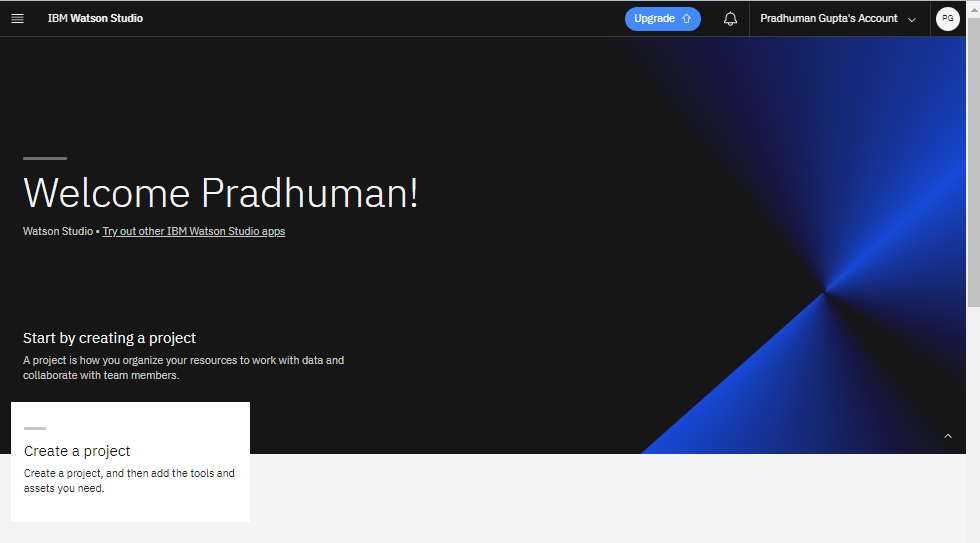
|  |
| --- |
| 1. EXPERIMENTAL INVESTIGATIONS |

* **IBM CLOUD RESOURCE LIST**

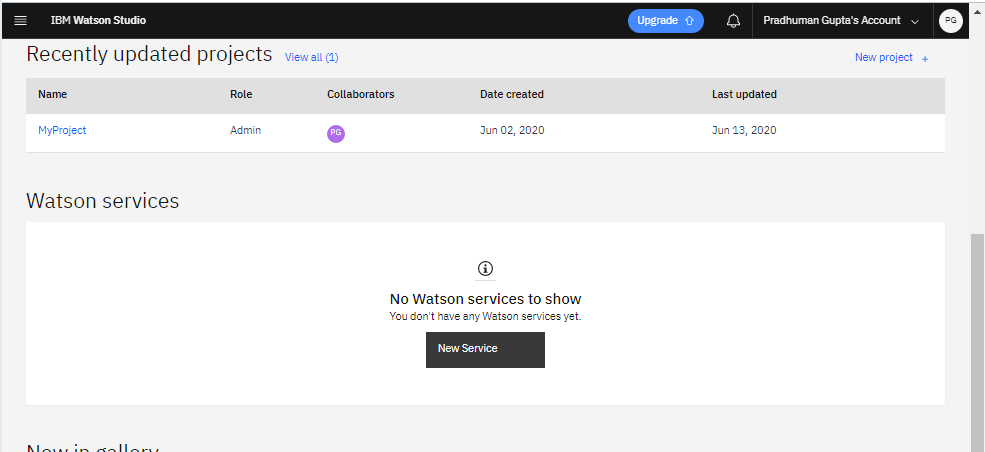


* **IBM WATSON STUDIO**

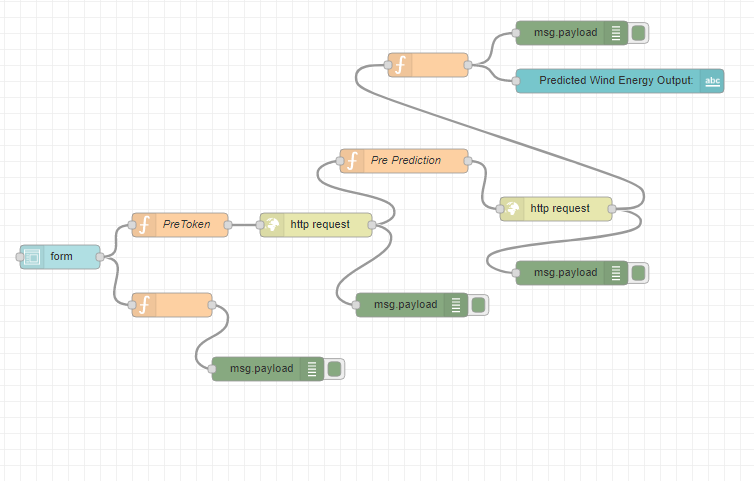


****

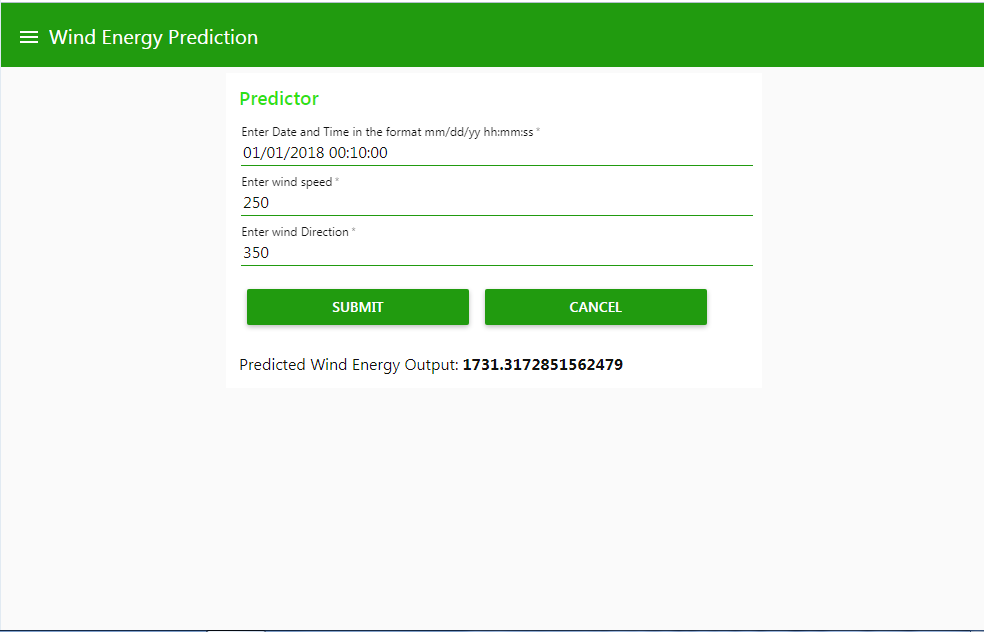
* **IBM CLOUD PROJECT DETAILS**



* **Node-RED FLOW**

****

* **LIFE EXPECTANCY PREDICTION UI**

****

|  |
| --- |
| 1. FLOW CHART |

**WATSON STUDIO**

**PREDICTING Wind Energy O/p**

**TRAINED DATA**

**MODEL**

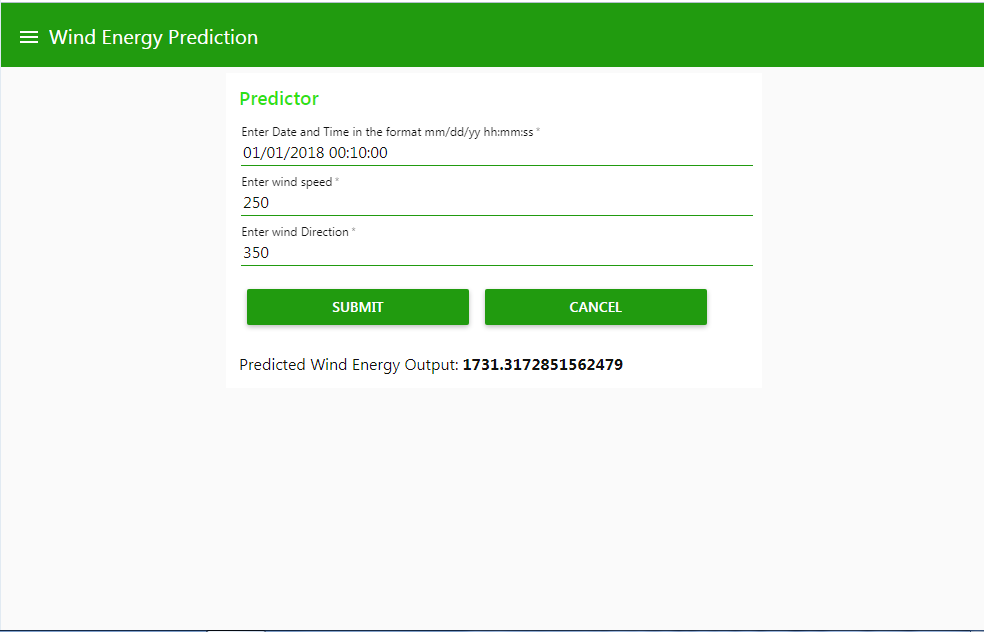
**APP UI**

**USER**

* The **USER** input all the Required Values in the App.
* The **Data** will Enter into **Watson** and the **Scoring\_Endpoint URL** matches with the **Deployed Model**.
* Then it Enters into the **Trained Data** and Predict **Wind Energy** Value.
* The Value predicted is opted into the **App** Screen.

|  |
| --- |
| 1. RESULT |

This is the **LIFE EXPECTANCY UI**.



|  |
| --- |
| 1. ADVANTAGES AND DISADVANTAGES |

|  |
| --- |
| ADVANTAGES |
| * Reduced Costs: This is a simple webpage and can be accessed by any citizen of a country to calculate Wind Energy Output Prediction at any Point of time and doesnot required any kind of payment neither for designing nor for using. * User Friendly Interface: This interface requires no background knowledge of how to use it. It’s a simple interface and only ask for required values and predict the output. |
| DISADVANTAGES |
| * Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value. * Average Prediction: The model predicts average or approximate value with 97.07% accuracy but not accurate value. |

|  |
| --- |
| 1. APPLICATIONS |

* It can be used to monitor Wind Energy Produced on any Point of time.
* It can be used to develop statistics for country development process.
* It is user friendly and can be used by anyone.

|  |
| --- |
| 1. CONCLUSION |

This user interface will be useful for the user to predict Wind Energy Produced by based on Date/Time , Wind Speed, Wind Direction.

|  |
| --- |
| 1. FUTURE SCOPE |

Future Scope of the Model can be:

* **Attractive UI**

It is a simple webpage only asking inputs and predict output. In future I have decided to make it more user friendly by providing some useful information about the country in the webpage itself so that user does not need to do any kind of prior research for the values.

* **Integrating with services such as speech recognition**

|  |
| --- |
| 1. BIBLIOGRAPHY |

* <https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L>
* <https://www.youtube.com/watch?v=Jtej3Y6uUng>
* <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service>
* <https://bookdown.org/caoying4work/watsonstudio-workshop/auto.html#add-asset-as-auto-ai>
* <https://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be>
* <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>

|  |
| --- |
| APPENDIX |

* **SOURCE CODE:**

**#Importing Libraries**

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**from** **sklearn.model\_selection** **import** train\_test\_split

**#Importing DataSet**

**import** **types**

**import** **pandas** **as** **pd**

**from** **botocore.client** **import** Config

**import** **ibm\_boto3**

**def** \_\_iter\_\_(self): **return** 0

*# @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\_2c4b212ecf374fa692c761ecd7f72bbb = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='N2YTIh93Yfvlw61IhV1f-i\_lmclF9FQpPBQCQ9xuwYPY',

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')

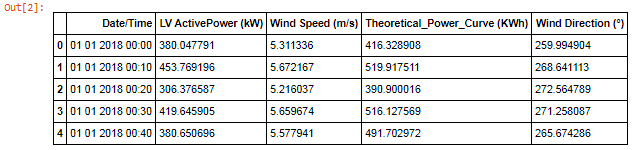
body = client\_2c4b212ecf374fa692c761ecd7f72bbb.get\_object(Bucket='myproject-donotdelete-pr-hzgogmf9qpe5kg',Key='Wind Data.csv')['Body']

*# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object*

**if** **not** hasattr(body, "\_\_iter\_\_"): body.\_\_iter\_\_ = types.MethodType( \_\_iter\_\_, body )

dataset = pd.read\_csv(body)

dataset.head()

****

*#dropping Theoretical power column cause it does'nt contribute to a good prediction*

dataset.drop('Theoretical\_Power\_Curve (KWh)',axis=1,inplace=**True**)

dataset.dtypes

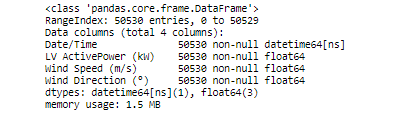


*# Convert Date/Time into Datetime*

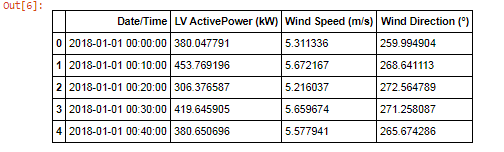
dataset['Date/Time']=pd.to\_datetime(dataset['Date/Time'])

data=dataset.copy()

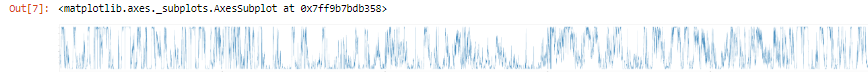
data.info()



data.head()



dataset['LV ActivePower (kW)'].plot(figsize=(350,20))



for feature in ['LV ActivePower (kW)','Wind Speed (m/s)','Wind Direction (°)']:

data=dataset.copy()

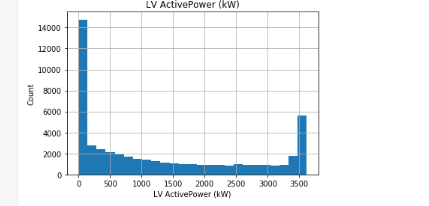
data[feature].hist(bins=25)

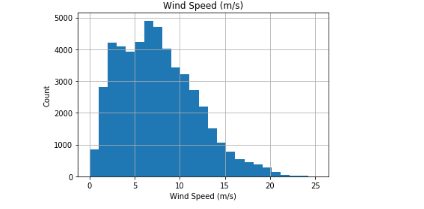
plt.xlabel(feature)

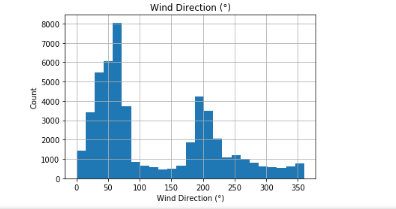
plt.ylabel("Count")

plt.title(feature)

plt.show()







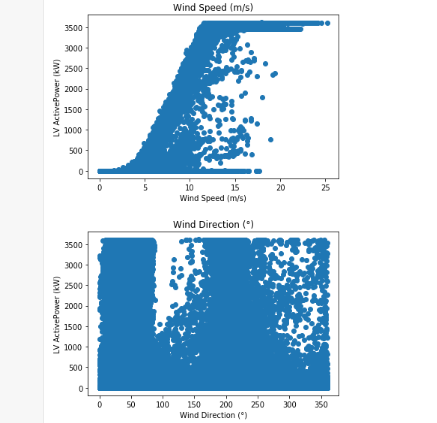
for feature in ['Wind Speed (m/s)','Wind Direction (°)']: data=dataset.copy()

plt.scatter(data[feature],data['LV ActivePower (kW)']) plt.xlabel(feature)

plt.ylabel('LV ActivePower (kW)')

plt.title(feature)

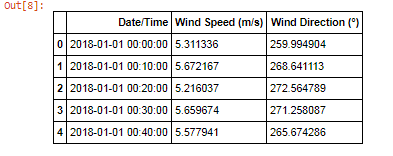
plt.show()



y=data['LV ActivePower (kW)'].values

data.drop('LV ActivePower (kW)',axis=1,inplace=**True**)

data.head()



print(data.shape)

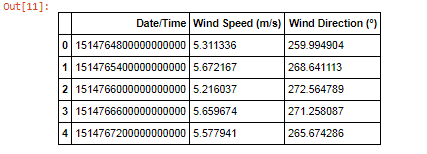
print(y.shape)



*#Converting Date/Time into milliseconds*

data['Date/Time']=pd.to\_numeric(data['Date/Time'])

data.head()



data.dtypes



*#splitting the dataset for Training and Testing*

X\_train,X\_test,y\_train,y\_test=train\_test\_split(data,y)

## Applying Random Forest Regression

**from** **sklearn.ensemble** **import** RandomForestRegressor

regressor=RandomForestRegressor(n\_estimators=10,random\_state=0)

**from** **sklearn.model\_selection** **import** cross\_val\_score

regressor.fit(X\_train,y\_train)

train\_score = cross\_val\_score(regressor,X\_train,y\_train,cv=5)

test\_score = cross\_val\_score(regressor,X\_test,y\_test,cv=5)

print("train score :",np.mean(train\_score))

print("test score :",np.mean(test\_score))



## Creating a Client with the ML Service Credentials

client = WatsonMachineLearningAPIClient(wml\_credentials)

model\_props = {

client.repository.ModelMetaNames.AUTHOR\_NAME : "Data Pirates",

client.repository.ModelMetaNames.AUTHOR\_EMAIL : "abhinavj65@gmail.com",

client.repository.ModelMetaNames.NAME : "windenergyprediction"

}

model\_artifact = client.repository.store\_model(regressor,meta\_props = model\_props)

model\_artifact

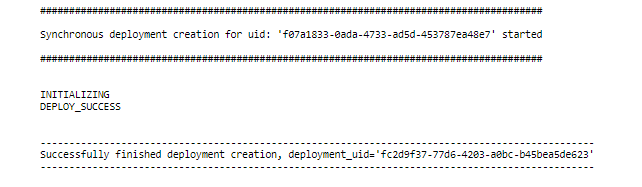
guid = client.repository.get\_model\_uid(model\_artifact)

guid

****

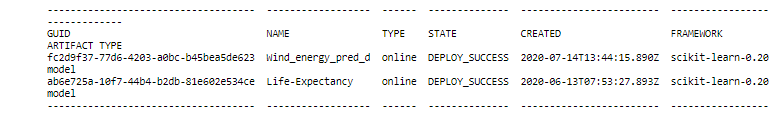
## Deployment Creation

deploy = client.deployments.create(guid,name = "Wind\_energy\_pred\_d")

****

*#List of deployments*

client.deployments.list()

****

## Scoring endpoint url

scoring\_url = client.deployments.get\_scoring\_url(deploy)

scoring\_url

