SMARTINTERNZ INTERNSHIP
PROJECT REPORT

TEAM ID: PNT2022TMID34159

PROJECT TITLE: Predicting the energy output of wind turbine based on weather condition

1. Introduction:

1.1 Project overview:

Category : Machine Learning
Skills Required : Python, Python for Data Analysis, Machine Learning, IBM Cloud

1.2 Purpose

Predicting the energy output of wind turbine based on weather conditions

2. Literature Survey :

2.1 Existing Problem

Wind energy plays increasing role in the supply of energy world-wide. The energy-output of a wind farm is highly dependent on the weather conditions present at its site. If the output is predicted more accurately, the energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, we do energy prediction based on weather data and analyse the important parameters as well as their correlation on the energy output.

2.2 References:

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

Weather data:
https://www.wunderground.com

Data Science:
https://www.youtube.com/watch?v=CmorAWRsCAw&list=PLeo1K3hjS3uuASpe-1LjfG5f14Bnozjwy

Climacell API:
https://www.climacell.co/weather-api/

Flask:
https://flask.palletsprojects.com/en/1.1.x/

Flutter
https://flutter.dev/

2.3 Problem Statement Definition:

The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	Admin	Produce accurate prediction of energy	I am unable to predict the energy	Weather condition	Stressed & confused
PS-2	User	Need the accurate prediction of energy	Admin did not give the correct reason	Change in climate condition	Troubled and worried

3. Ideation and Proposed Solution:

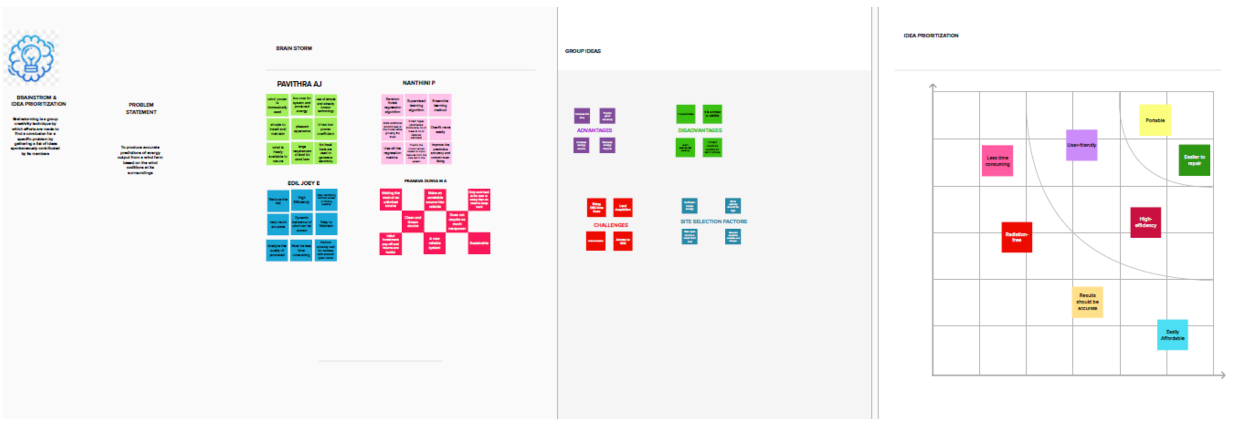
3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

3.2 Ideation and Brainstorming:

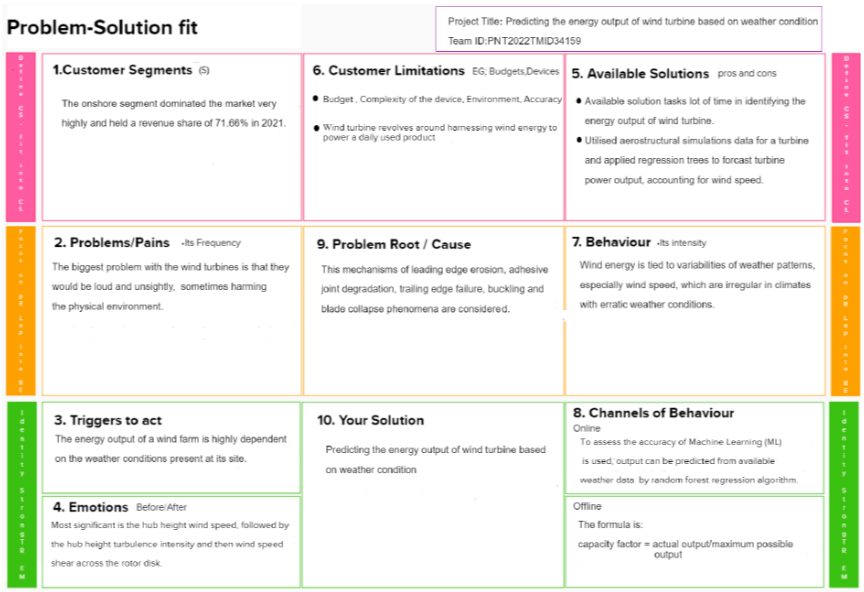
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.



3.3 Proposed Solution:

Our aim is to map weather data to energy production. We wish to show that even data that is publicly available for weather stations close to wind farms can be used to give a good prediction of the energy output. Furthermore, we examine the impact of different weather conditions on the energy output of the wind farms. We are building an IBM Watson Auto AI Machine Learning technique to predict the energy output of wind turbine. We deploy the model. On IBM cloud to get scoring end point. It can be used as API in mobile app or web app building. We are developing a web application which is built using node red service. We use the scoring end point to give user input values to the deployed model. The model prediction is then showcased on User Interface to predict the energy output of wind turbine.

3.4 Problem Solution fit:



4. Requirement Analysis:

4.1 Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Input	Location of the user must be ON state on the device.
FR-4	Weather Condtion	Based on location the Weather condition data is gathered through any satellites or resourse like API
FR-5	Regression tree method	To get more reliable output these methods are used.
FR-6	Modeling the data	This is used to Genetic programming will helps to get more accurate result.
FR-7	Wind Energy Prediction	A synergeic neural network based model is used.
FR-8	Energy Output	The formula is *capacity factor=actual output/maximum possible output* as ***kWh/yr***

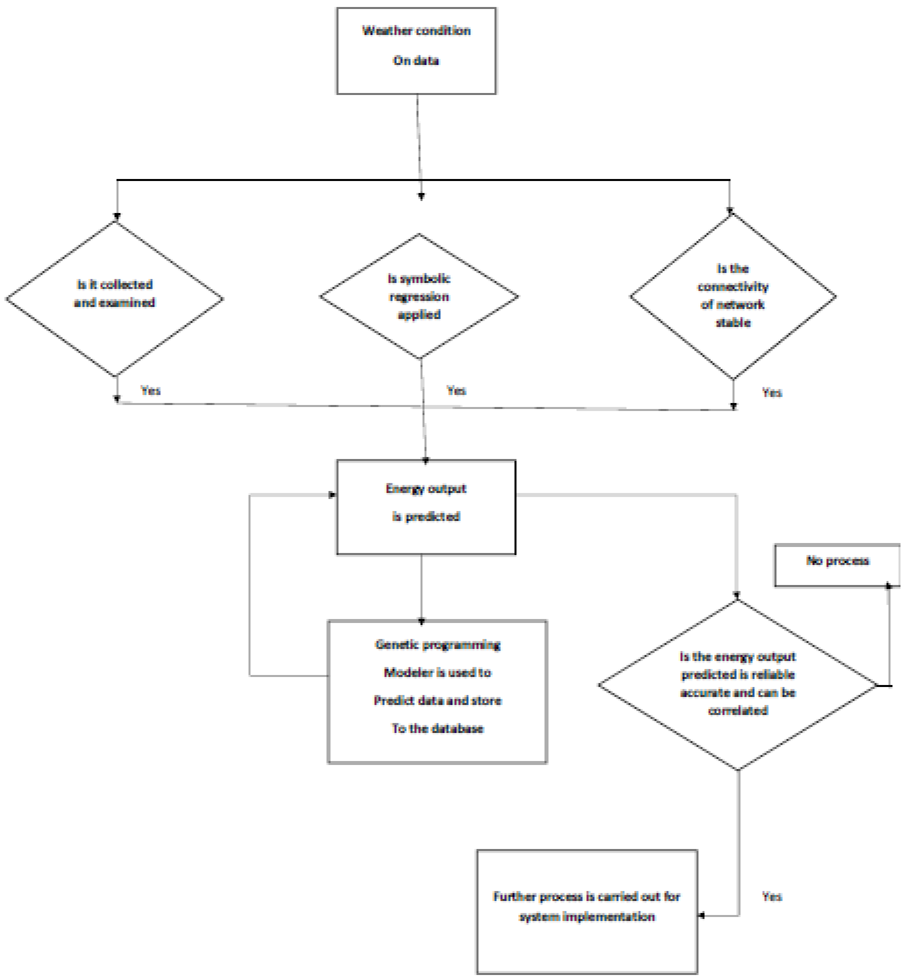
4.2 Non-Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• It must be user friendly for all medium of people. • More Effective. • Less Data consumption.
NFR-2	**Security**	• Secure Software Development • Forms authentication must be deployed. • Encryption ,key Management ,Firewall and Router Management.
NFR-3	**Reliability**	Wind energy is reliable because it Highly securable, unlimited and effective integrated data.
NFR-4	**Performance**	There we use Machine Learning techniques are combined so, the system performs well under all critical circumstance thus provide the user a well satisfied interface
NFR-5	**Availability**	The most required is a device with good internet connection,they are globally available to all user across the world.
NFR-6	**Scalability**	It will perform well under an increased or expanding workload as huge storage data and to retrieval data.

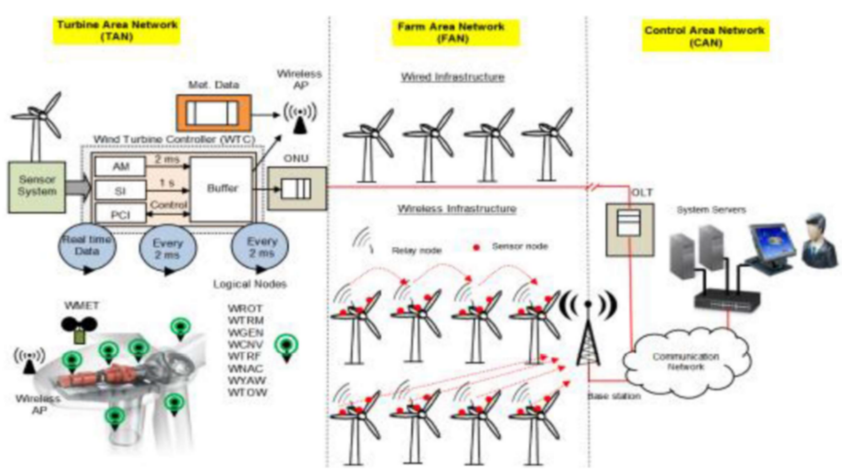
5. Project Design:

5.1 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Installation	USN-1	I can install the energy predictor as an industries safely	I can do it myself	High	Sprint-1
Customer	Handling of device	USN-2	The device should be handled safely	I will handle it	High	Sprint-2
Customer	Safety	USN-3	The device should not have any contact with excessive heat or cold beyond the limit	I will ensure that	High	Sprint-3
Customer	Power connectivity	USN-4	The power should be given at a perfect quantity and should not be overloaded	I can assure that	Medium	Sprint-4
Customer	Internet connectivity	USN-5	Both wired and wireless network connectivity should be at any time	I will ensure that	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

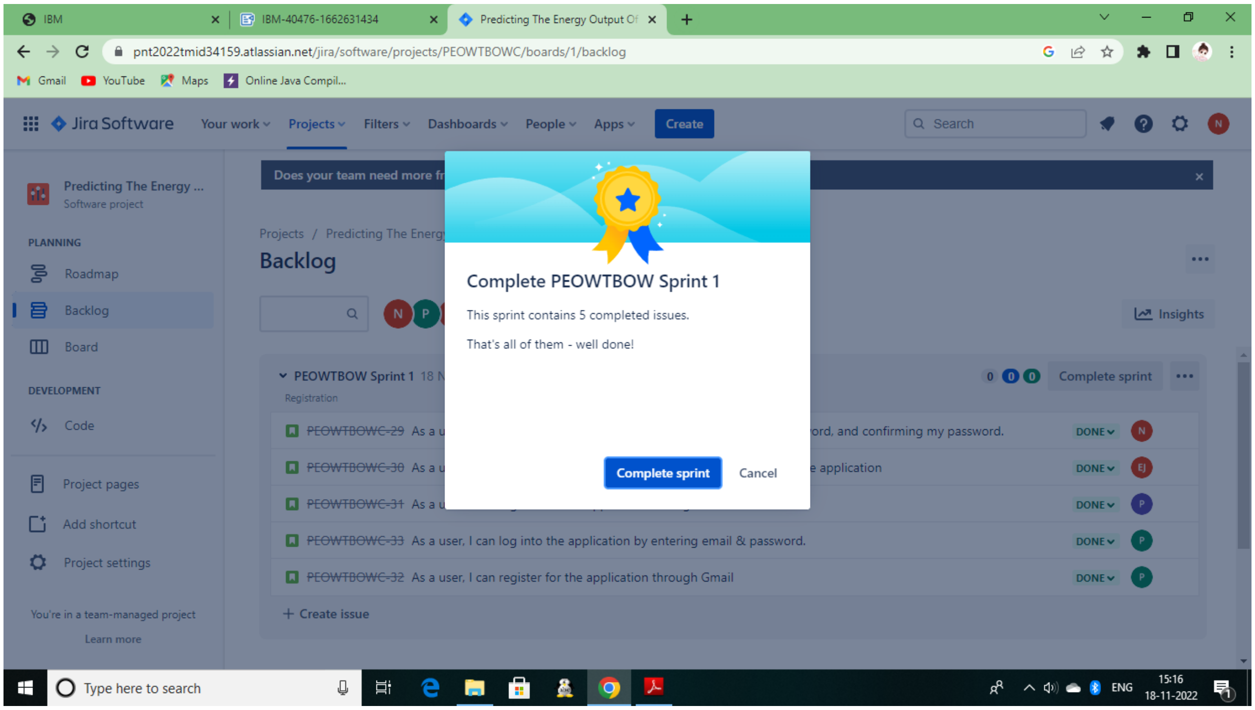
6.1 Sprint Planning & Estimation

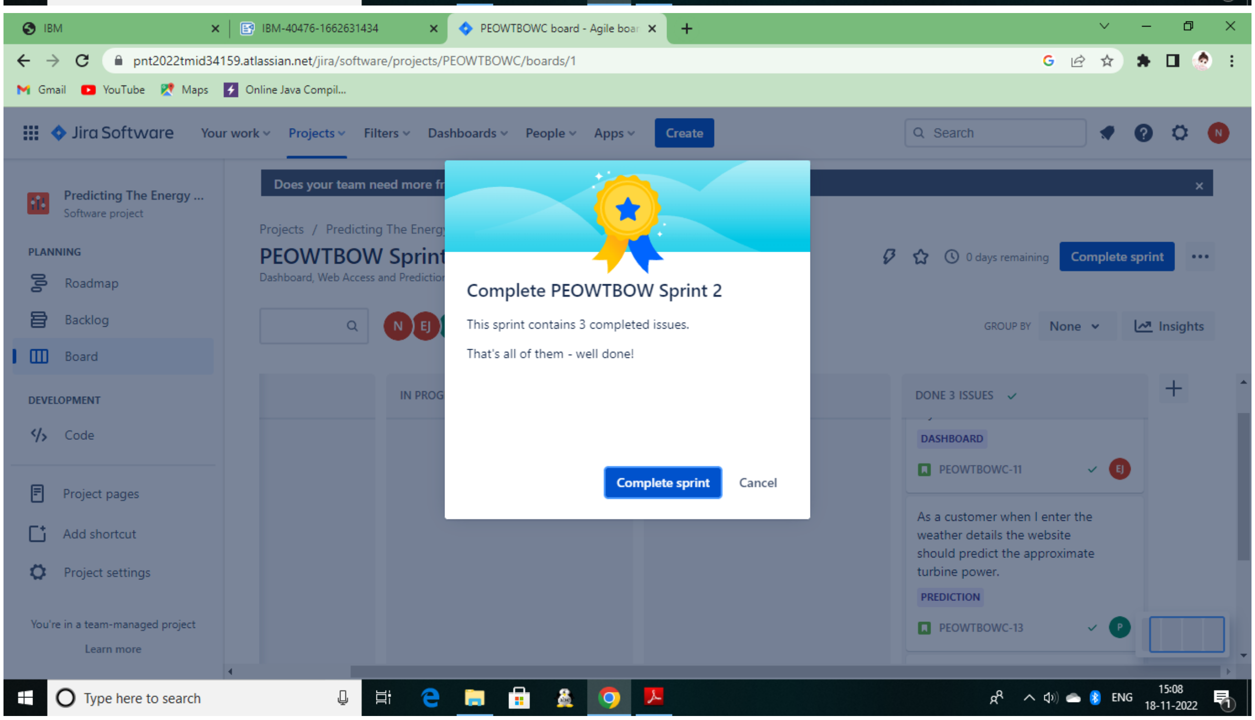
Sprint	Milestone
Sprint 1	1. User Registers into the application through entering Email Id Password and Re enter Password for confirmation. 2. User Receives a confirmation mail for their registered Email. 3. User can also register to the application through Mobile number. 4. User logs in into the website using Email Id password or through Gmail
Sprint 2	1. User can access the dashboard 2. User can enter the required details on weather conditions and get the desired turbine power output based on model's prediction
Sprint 3	1. Application should store the predictions, and these predictions can be used for future analysis. 2. The data stored should be secure.
Sprint 4	1. Administrator should properly maintain the website and update it when required.

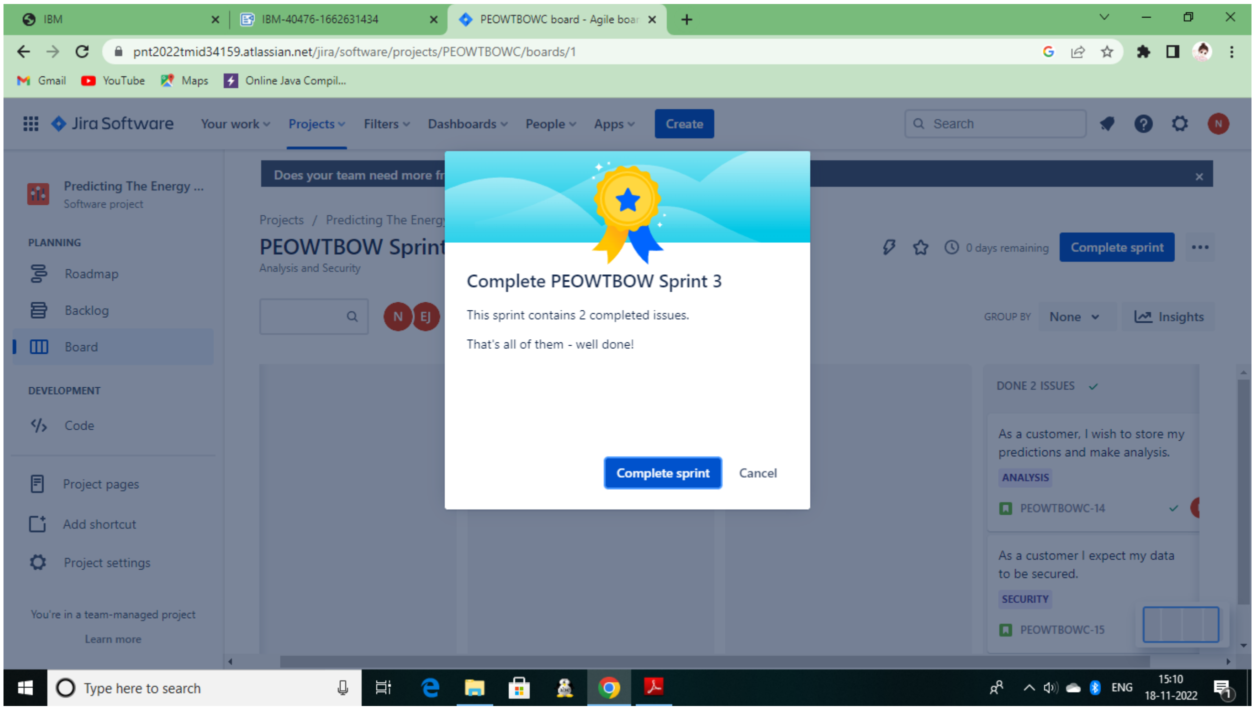
6.2 Sprint Delivery Schedule

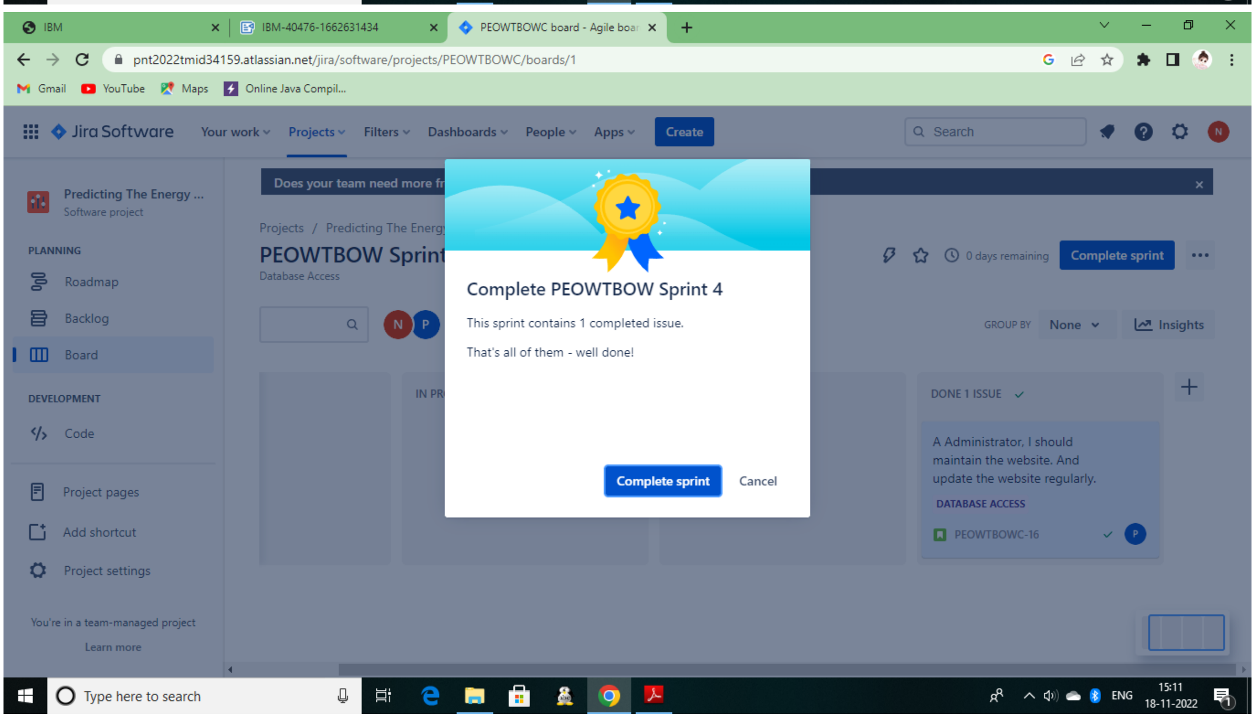
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint -3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

6.3 Reports from JIRA



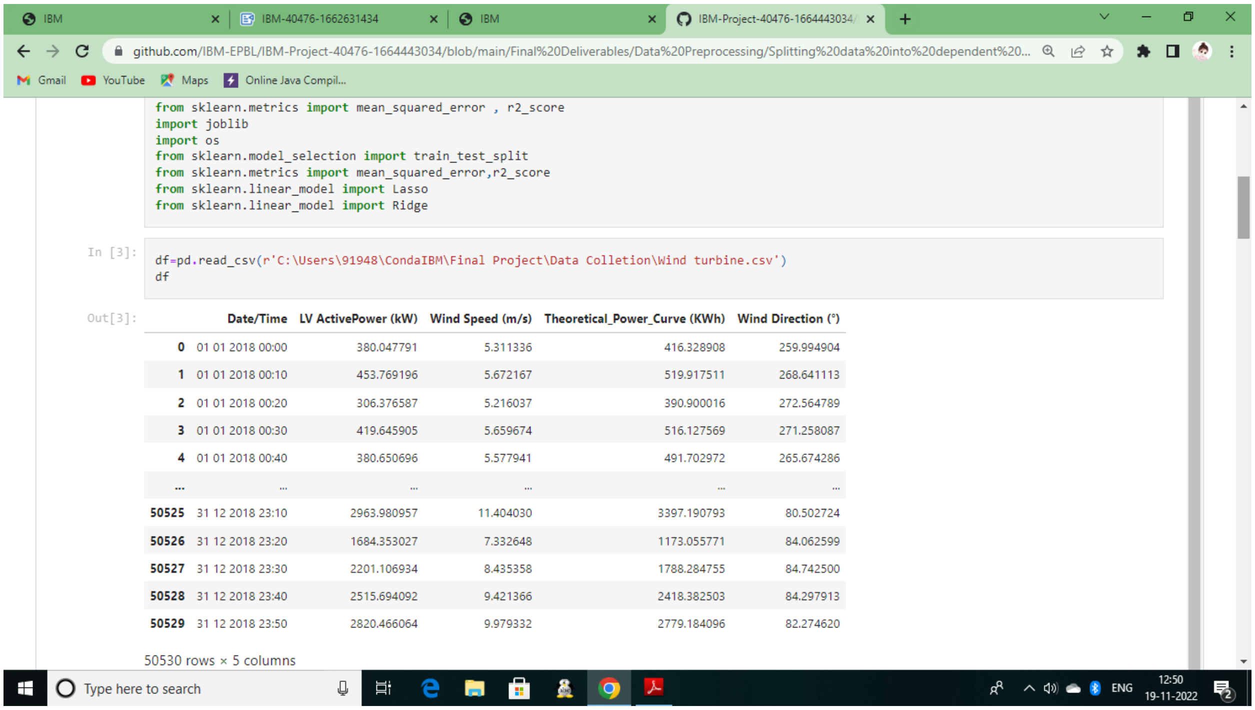






7. CODING & SOLUTIONING

7.1 Feature 1




```
from sklearn.metrics import mean_squared_error , r2_score
import joblib
import os
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
```

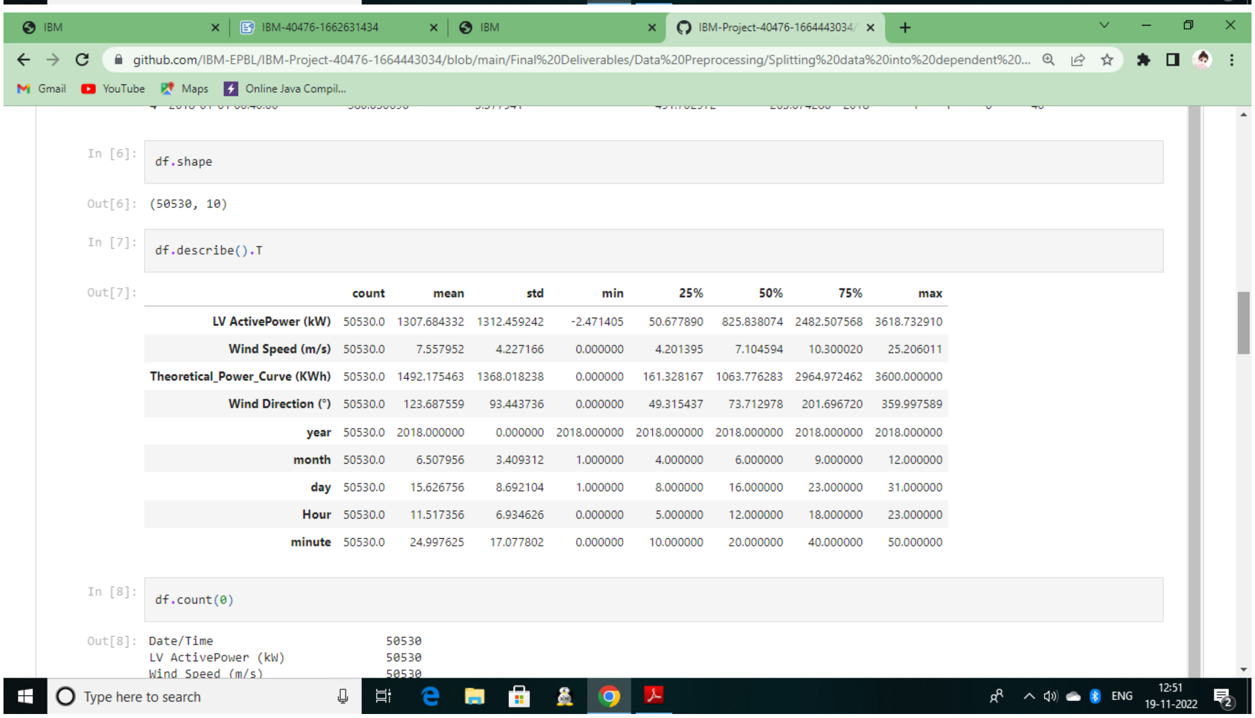
In [3]:

```
df=pd.read_csv(r'C:\Users\91948\CondaIBM\Final Project\Data Colletion\Wind turbine.csv')
df
```

Out[3]:

	Date/Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
4	01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286
...	...	...	...	...	...
50525	31 12 2018 23:10	2963.980957	11.404030	3397.190793	80.502724
50526	31 12 2018 23:20	1684.353027	7.332648	1173.055771	84.062599
50527	31 12 2018 23:30	2201.106934	8.435358	1788.284755	84.742500
50528	31 12 2018 23:40	2515.694092	9.421366	2418.382503	84.297913
50529	31 12 2018 23:50	2820.466064	9.979332	2779.184096	82.274620

50530 rows × 5 columns



In [6]:

```
df.shape
```

Out[6]: (50530, 10)

In [7]:

```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
LV ActivePower (kW)	50530.0	1307.684332	1312.459242	-2.471405	50.677890	825.838074	2482.507568	3618.732910
Wind Speed (m/s)	50530.0	7.557952	4.227166	0.000000	4.201395	7.104594	10.300020	25.206011
Theoretical_Power_Curve (KWh)	50530.0	1492.175463	1368.018238	0.000000	161.328167	1063.776283	2964.972462	3600.000000
Wind Direction (°)	50530.0	123.687559	93.443736	0.000000	49.315437	73.712978	201.696720	359.997589
year	50530.0	2018.000000	0.000000	2018.000000	2018.000000	2018.000000	2018.000000	2018.000000
month	50530.0	6.507956	3.409312	1.000000	4.000000	6.000000	9.000000	12.000000
day	50530.0	15.626756	8.692104	1.000000	8.000000	16.000000	23.000000	31.000000
Hour	50530.0	11.517356	6.934626	0.000000	5.000000	12.000000	18.000000	23.000000
minute	50530.0	24.997625	17.077802	0.000000	10.000000	20.000000	40.000000	50.000000

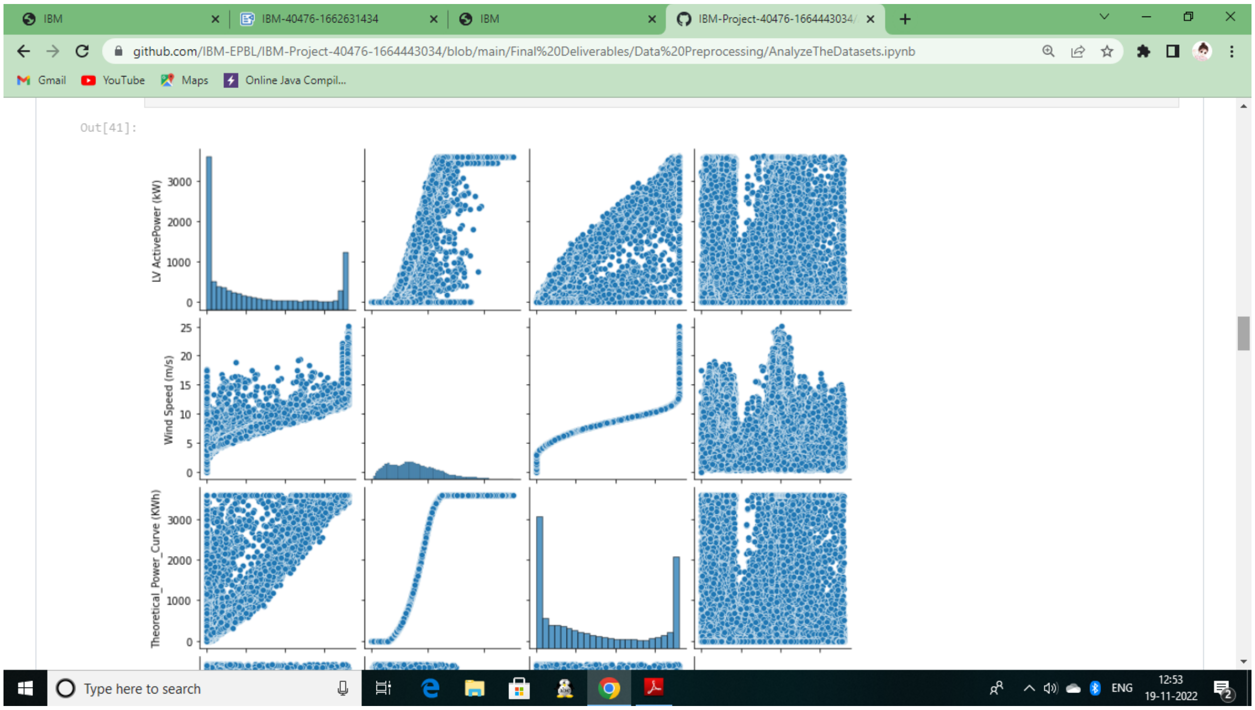
In [8]:

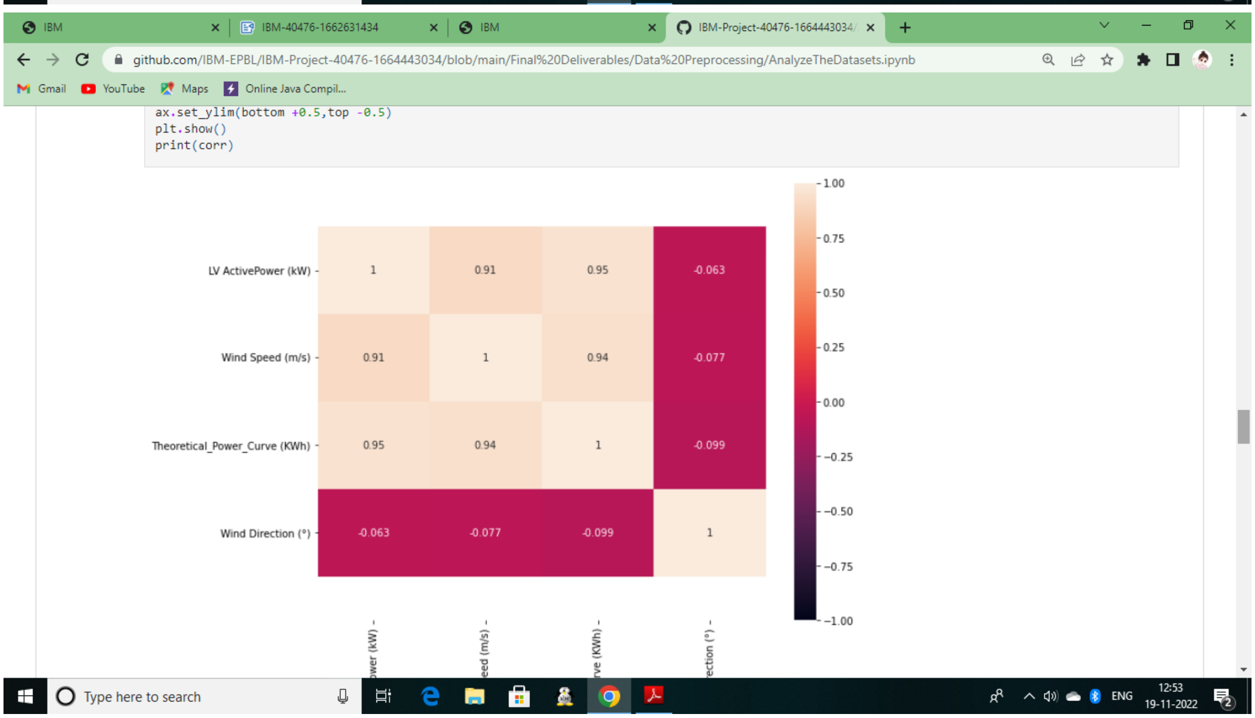
```
df.count(0)
```

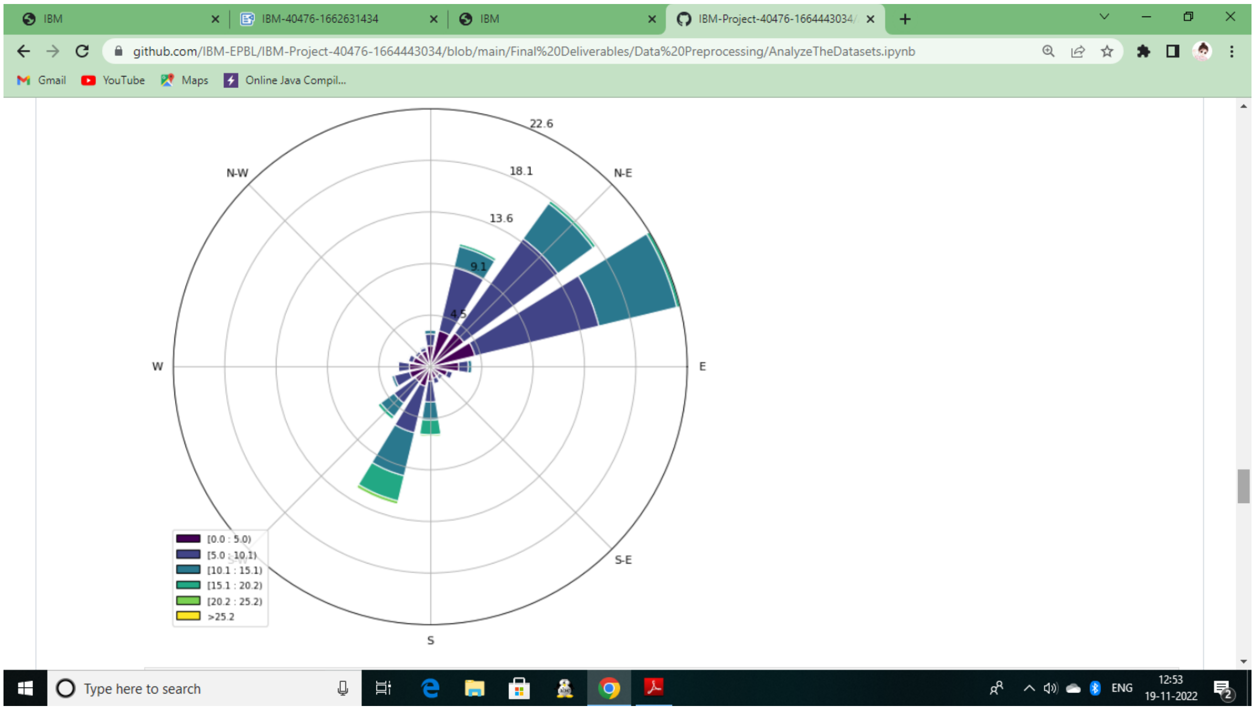
Out[8]:

```
Date/Time                  50530
LV ActivePower (kW)        50530
Wind Speed (m/s)           50530
```

7.2 Feature 2

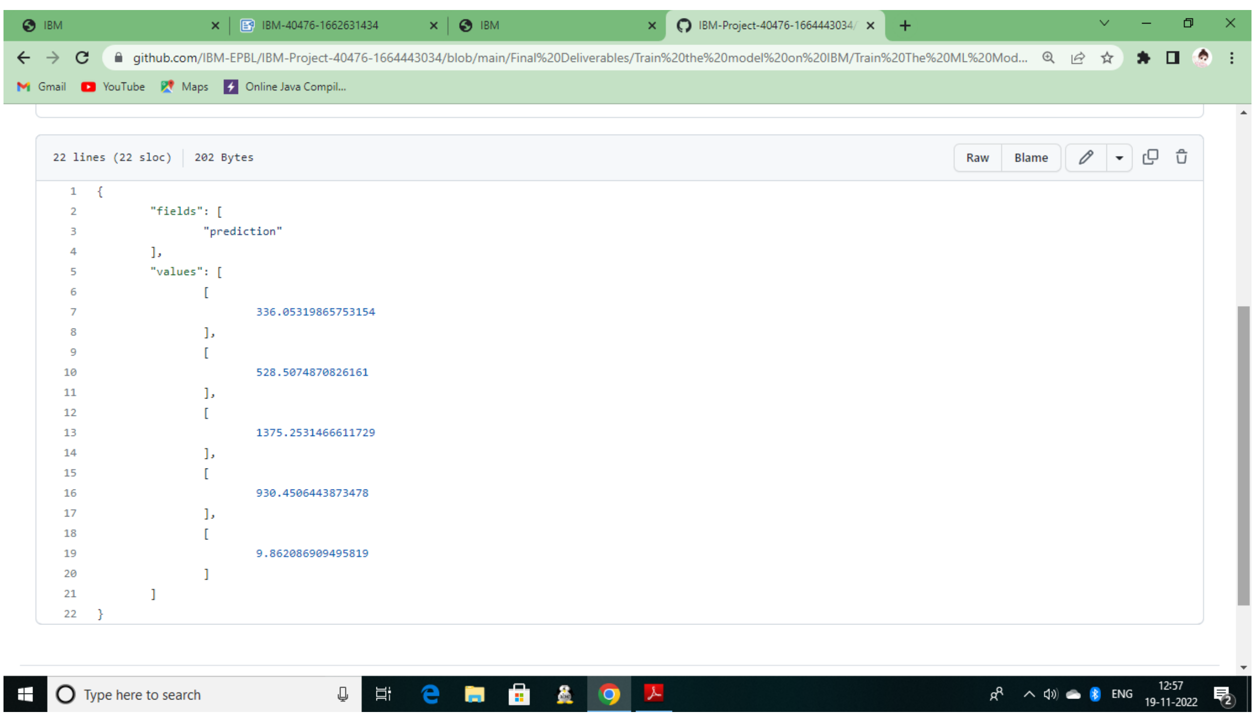

IBM
IBM-40476-1662631434
IBM
IBM-Project-40476-1664443034/
github.com/IBM-EPBL/IBM-Project-40476-1664443034/blob/main/Final%20Deliverables/Data%20Preprocessing/AnalyzeTheDatasets.ipynb
Gmail
YouTube
Maps
Online Java Compil...
Out[41]:
LV ActivePower (kW)
Wind Speed (m/s)
Theoretical_Power_Curve (KWh)
3000
2000
1000
0
25
20
15
10
5
0
Type here to search
ENG
12:53
19-11-2022


IBM
IBM-40476-1662631434
IBM
IBM-Project-40476-1664443034/
github.com/IBM-EPBL/IBM-Project-40476-1664443034/blob/main/Final%20Deliverables/Data%20Preprocessing/AnalyzeTheDatasets.ipynb
Gmail
YouTube
Maps
Online Java Compil...
ax.set_ylim(bottom +0.5,top -0.5)
plt.show()
print(corr)
LV ActivePower (kW)
Wind Speed (m/s)
Theoretical_Power_Curve (KWh)
Wind Direction (°)
1
0.91
0.95
-0.063
0.91
1
0.94
-0.077
0.95
0.94
1
-0.099
-0.063
-0.077
-0.099
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Type here to search
ENG
12:53
19-11-2022

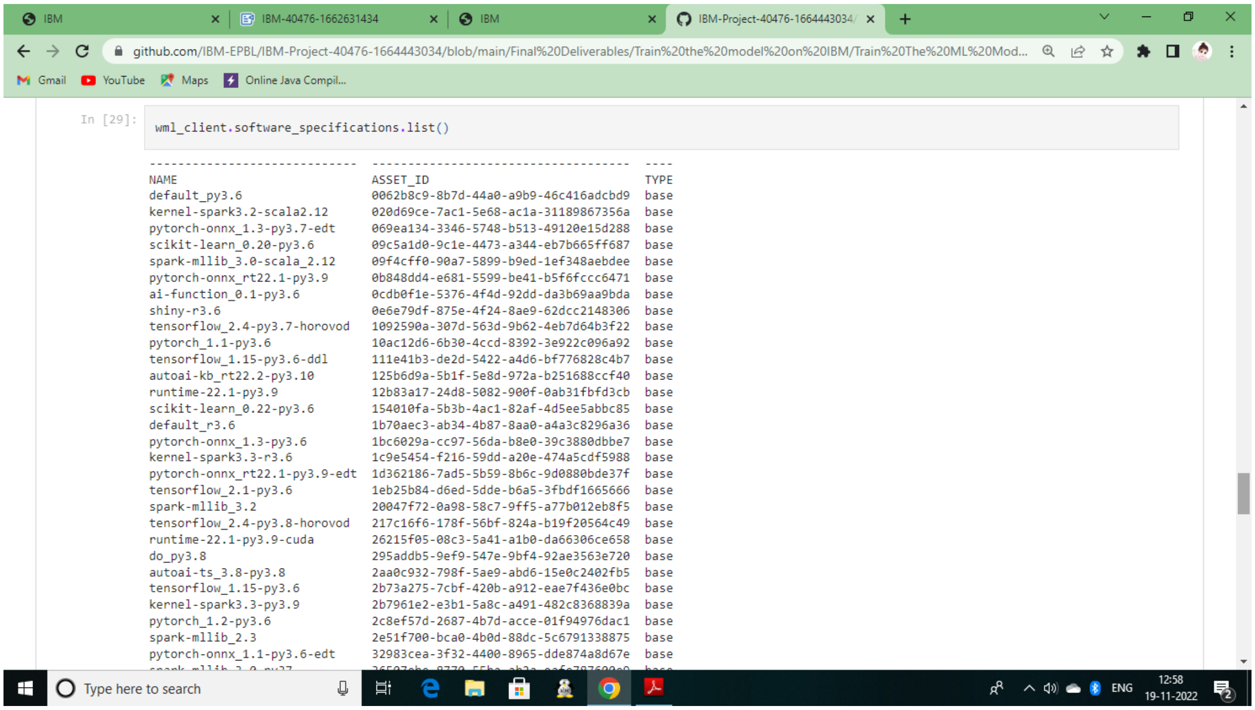


8. TESTING

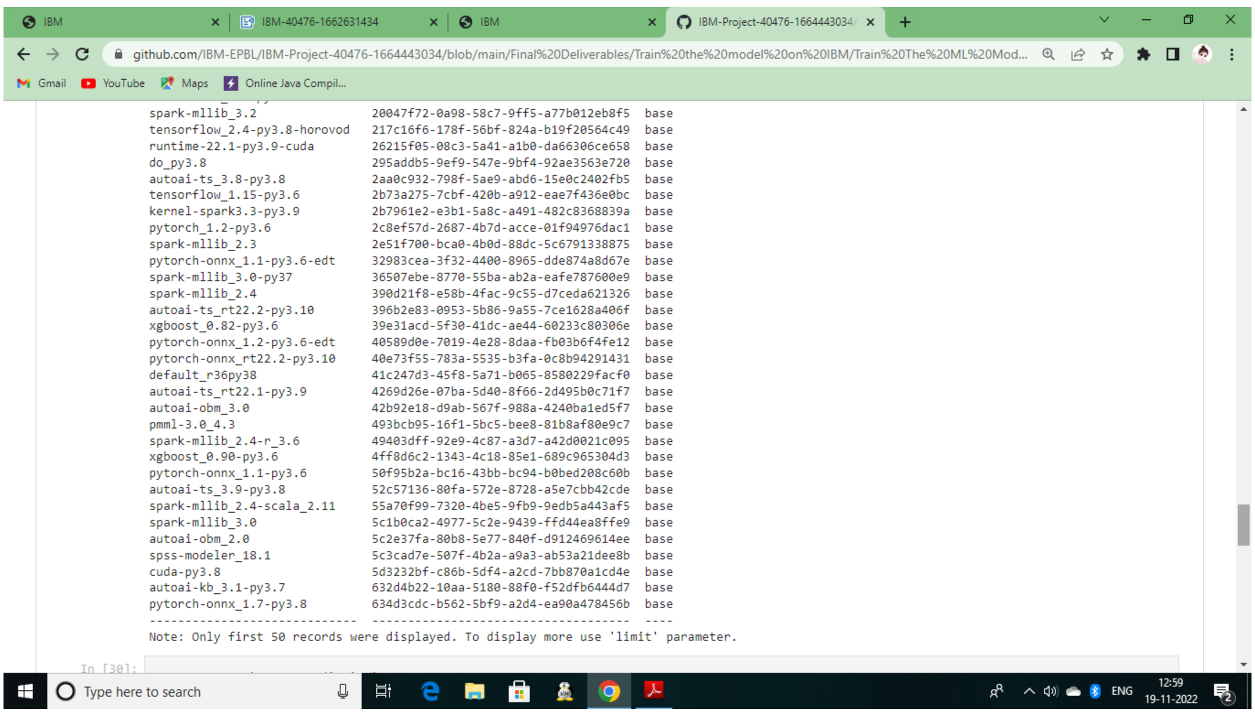
8.1 Test Cases



8.2 User Acceptance Testing



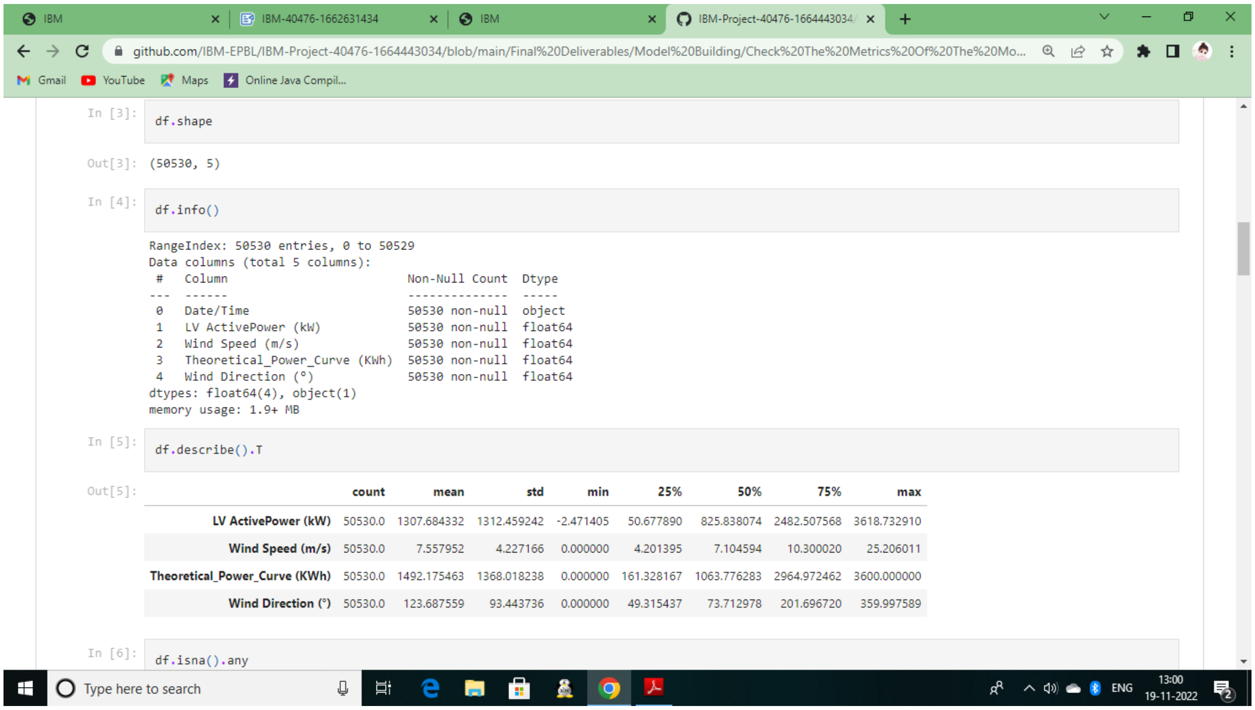
```
In [29]: wml_client.software_specifications.list()

----------------------------  ------------------------------------  ----
NAME                          ASSET_ID                              TYPE
default_py3.6                 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12     020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt    069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6       09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12    09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9     0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6         0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                    0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod  1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6             10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl     111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10       125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9            12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6       154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                  1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6        1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6          1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6          1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2               20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod  217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda       26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                      295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8           2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6         2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9         2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6             2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3               2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt    32983cea-3f32-4400-8965-dde874a8d67e  base
```

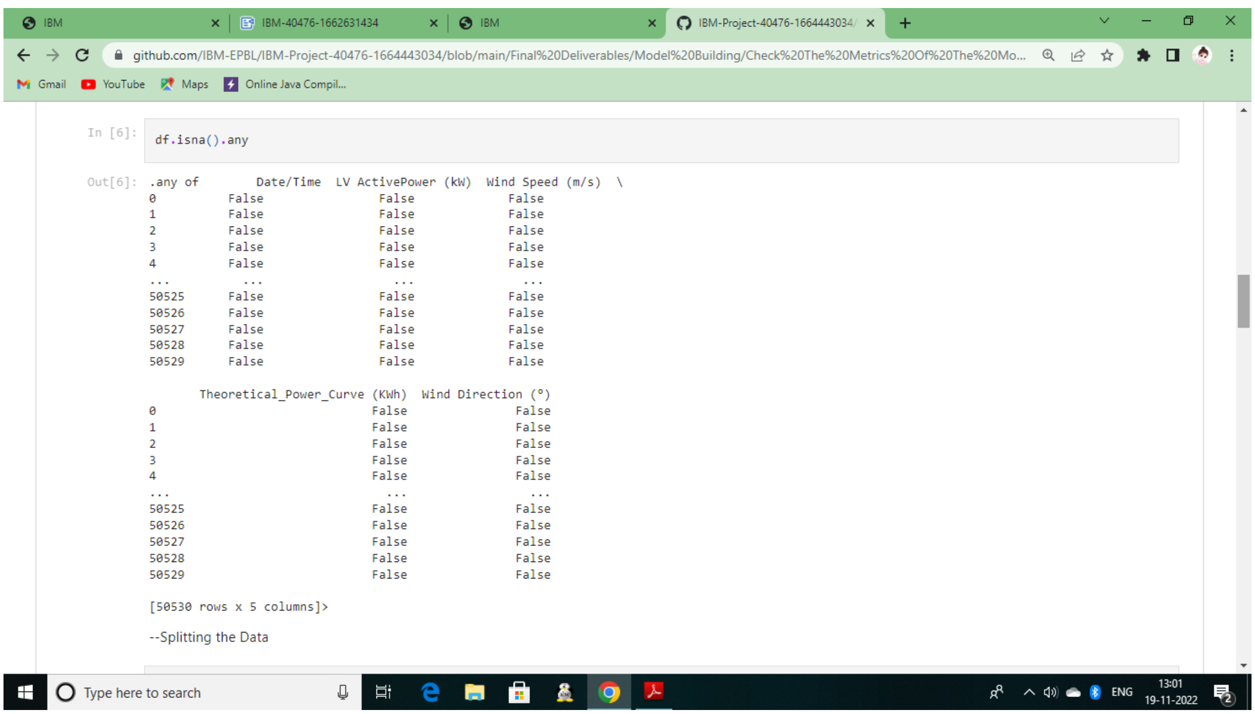


```
spark-mllib_3.2               20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod  217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda       26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                      295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8           2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6         2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9         2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6             2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3               2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt    32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37          36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4               390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10       396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6            39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt    40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10    40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38               41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9        4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                  493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6         49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6            4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6        50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8           52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11    55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0               5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1             5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                    5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7           632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8        634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

9. RESULTS

9.1 Performance Metrics





10. Advantages and Disadvantages:

10.1 Advantages:

1. Weather Underground Services provide very accurate Historical

Weather Data which increased the accuracy of model.

2. Mobile App is more convenient to use rather than web apps.
3. On giving location permissions, app can accurately predict power output at your live location.

10.2 Disadvatages:

1. Weather API is paid and the free version provide limited API requests per day.
2. Android App can't be deployed on IBM Cloud.
3. No free server available on IBM Cloud for deploying Backend.

11. Conclusion:

We started with the aim of improving the predictions of power generated using wind energy and we have achieved that using LSTM as machine learning model and performing model optimization on it. We have also observed that if the wind speed is less than 4 m/s the power generated by the system is zero. LSTM is not able to learn this pattern as this is not the part which it can understand in time series analysis. So, if a hybrid new model is created which can work as the combination of Decision Tree/Random Forest and LSTM we can improve upon these results as well.

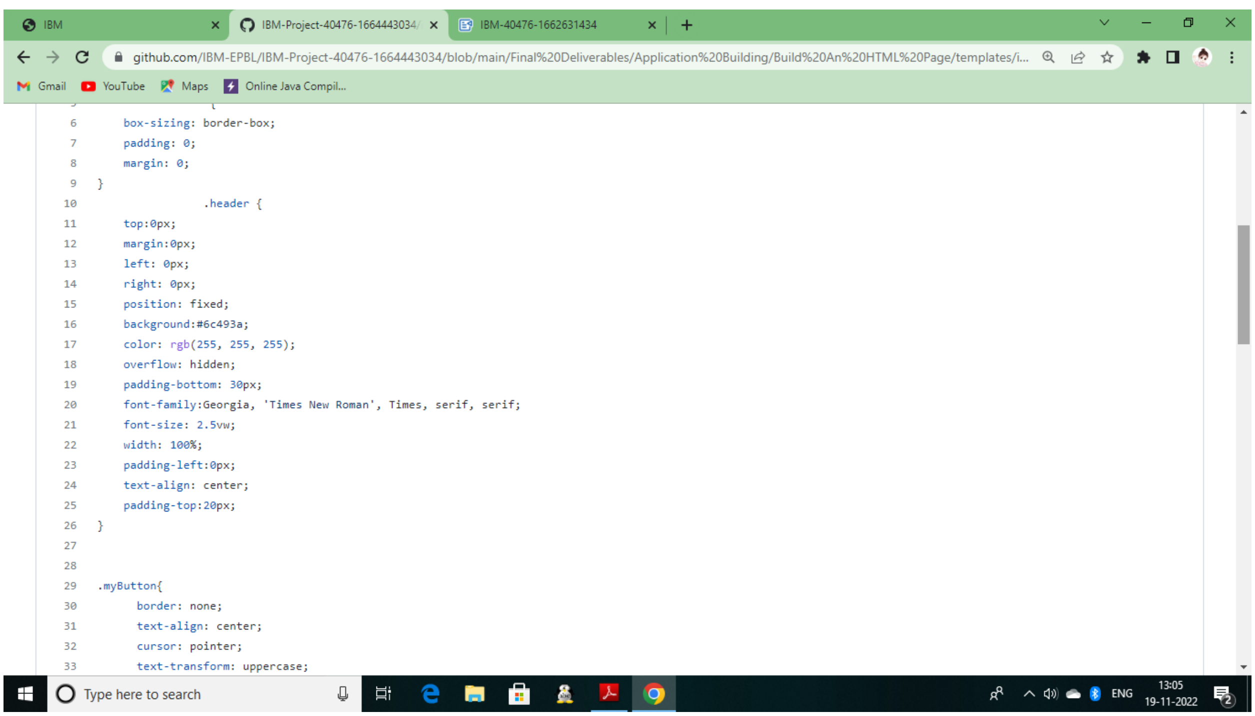
12. Future Scope:

Most wind power forecasting models study ‘regular’ wind conditions. The EU funded project called ‘Safewind’ aims to improve wind power prediction over challenging and extreme weather periods and at different temporal and spatial scales. Development activities are on-going to reduce error in the wind power prediction, to improve regionalized wind power forecasting for on - shore wind farms and to derive methods for wind power prediction for offshore wind farms. It is possible that use of ensemble and combined weather prediction methods together may enhance forecasting.

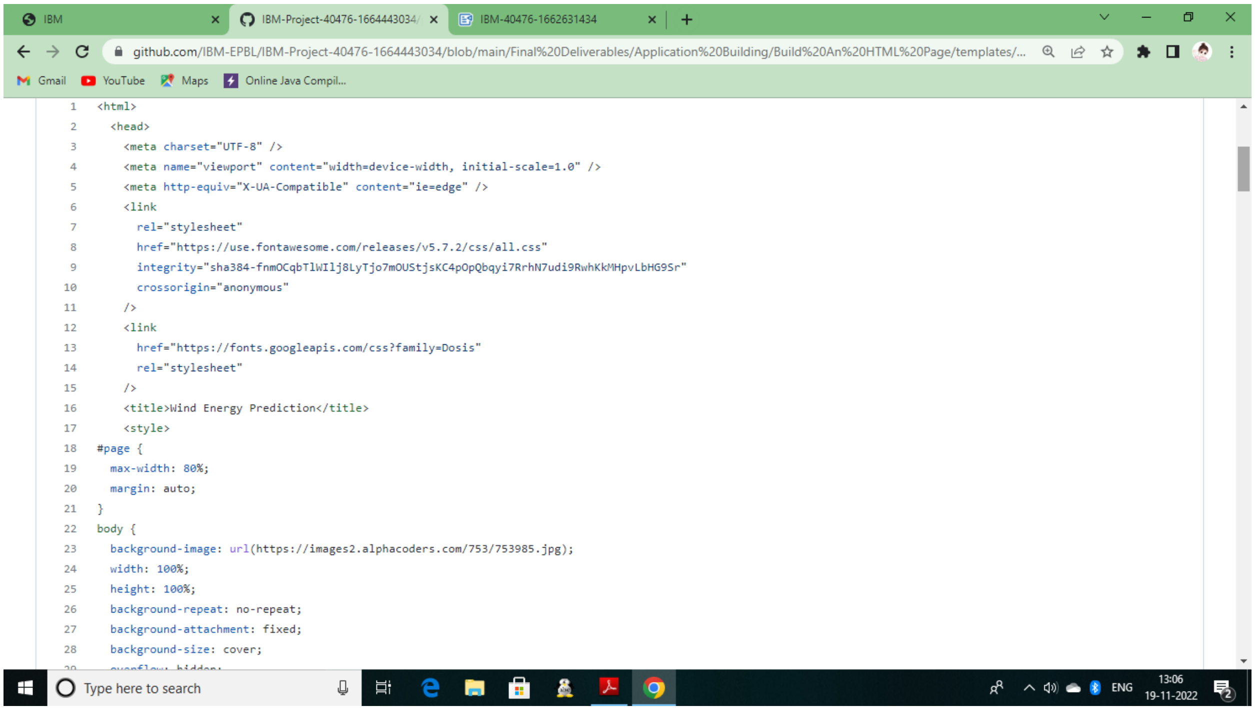
If the error in wind power forecasting and prediction is reduced then electricity markets can trade with more certainty. Contract errors as a function of time in electricity markets can be as high as 39% for a forecasting lead time of 4 h. Gubina et al. present a new tool called the WILMAR and ANEMOS scheduling Methodology (WALT) to reduce the number of thermal generators on stand by or in reserve using the probability of generation outages and load shedding are system reliability criteria instead of generation adequacy based solely on generation outage. The wind and load forecast errors are modelled using a Gaussian stochastic variable approach. However, in another study it was found that the prediction errors do not satisfy the KolmogoroveSmirnov test for normal distribution. In Ramìrez and Carta , it was shown that, the use of autocorrelated (and thus not independent) successive hourly mean wind speeds, though invalidating all of the usual statistical tests, has no appreciable effect on the shape of the pdf estimated from the data.

13. APPENDIX

Source Code



```
    box-sizing: border-box;
    padding: 0;
    margin: 0;
}
                .header {
    top:0px;
    margin:0px;
    left: 0px;
    right: 0px;
    position: fixed;
    background:#6c493a;
    color: rgb(255, 255, 255);
    overflow: hidden;
    padding-bottom: 30px;
    font-family:Georgia, 'Times New Roman', Times, serif, serif;
    font-size: 2.5vw;
    width: 100%;
    padding-left:0px;
    text-align: center;
    padding-top:20px;
}


.myButton{
      border: none;
      text-align: center;
      cursor: pointer;
      text-transform: uppercase;
```



GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-40476-1664443034

https://youtu.be/m8Y3ovRcnsA