PV Generation Forecasting Based On Cloud Detection Calculation Via Camera And Sensor Network

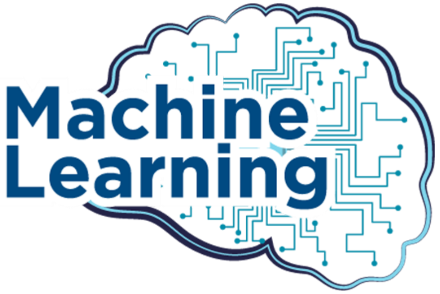
Mehmet Fatih Altun

End of Semester Presentations for SDP 1

Project Description

- Solar panels PV Generation Forecasting

- To Create a Forecasting Algorithms
  - Tried to Develop Algorithms
- A software Project
  - Python



Dataset Before Preprocessing

Index	Day of Year	Year	Month	Day	First Hour of Period	Is Daylight	Distance to Solar Noon	Average Temperature (Day)	Wind Directi	ge Wind Speed	Sky Cover	Visibility	Relative Humidity	Average Wind Speed (Period)	Barometric Pressure	Power Generated
0	245	2008	9	1	1	False	0.859897	69	28	7.5	0	10	75	8	29.82	0
1	245	2008	9	1	4	False	0.628535	69	28	7.5	0	10	77	5	29.85	0
2	245	2008	9	1	7	True	0.397172	69	28	7.5	0	10	70	0	29.89	5418
3	245	2008	9	1	10	True	0.16581	69	28	7.5	0	10	33	0	29.91	25477
4	245	2008	9	1	13	True	0.0655527	69	28	7.5	0	10	21	3	29.89	30069
5	245	2008	9	1	16	True	0.296915	69	28	7.5	0	10	20	23	29.85	16280
6	245	2008	9	1	19	True	0.528278	69	28	7.5	0	10	36	15	29.83	515
7	245	2008	9	1	22	False	0.75964	69	28	7.5	0	10	49	6	29.86	0
8	246	2008	9	2	1	False	0.862113	72	29	6.8	0	10	67	6	29.86	0
9	246	2008	9	2	4	False	0.630155	72	29	6.8	0	10	49	0	29.87	0
10	246	2008	9	2	7	True	0.398196	72	29	6.8	0	10	54	0	29.9	4939
11	246	2008	9	2	10	True	0.166237	72	29	6.8	0	10	64	0	29.92	24335
12	246	2008	9	2	13	True	0.0657216	72	29	6.8	0	10	23	9	29.88	29025
13	246	2008	9	2	16	True	0.29768	72	29	6.8	0	10	30	18	29.84	15408
14	246	2008	9	2	19	True	0.529639	72	29	6.8	0	10	65	11	29.84	491
15	246	2008	9	2	22	False	0.761598	72	29	6.8	0	10	75	5	29.85	0
16	247	2008	9	3	1	False	0.865459	73	29	7.9	0	10	72	6	29.84	0
17	247	2008	9	3	4	False	0.6326	73	29	7.9	0	10	78	6	29.86	0
18	247	2008	9	3	7	True	0.399741	73	29	7.9	0	10	63	0	29.88	4854
19	247	2008	9	3	10	True	0.166882	73	29	7.9	0	10	69	3	29.88	23855
20	247	2008	9	3	13	True	0.0659767	73	29	7.9	0	10	33	10	29.84	28339
21	247	2008	9	3	16	True	0.298836	73	29	7.9	0	10	27	20	29.81	15308
22	247	2008	9	3	19	True	0.531695	73	29	7.9	0	10	53	14	29.81	455
23	247	2008	9	3	22	False	0.764554	73	29	7.9	0	10	70	6	29.85	0
24	248	2008	9	4	1	False	0.867704	76	30	6.9	0	10	73	0	29.83	0

2920 rows
16 variables

A solar power system installed in Berkeley, CA.

Datasets After Preprocessing

independent variable

Index	Is Daylight	Distance to Solar Noon	Average Temperature (Day)	Average Wind Direction (Day)	Average Wind Speed (Day)	Sky Cover	Visibility	Relative Humidity	Average Wind Speed (Period)	Average Barometric Pressure (Period)
0	0	0.859897	69	28	7.5	0	10	75	8	29.82
1	0	0.628535	69	28	7.5	0	10	77	5	29.85
2	1	0.397172	69	28	7.5	0	10	70	0	29.89
3	1	0.16581	69	28	7.5	0	10	33	0	29.91
4	1	0.0655527	69	28	7.5	0	10	21	3	29.89
5	1	0.296915	69	28	7.5	0	10	20	23	29.85
6	1	0.528278	69	28	7.5	0	10	36	15	29.83
7	0	0.75964	69	28	7.5	0	10	49	6	29.86
8	0	0.862113	72	29	6.8	0	10	67	6	29.86
9	0	0.630155	72	29	6.8	0	10	49	0	29.87
10	1	0.398196	72	29	6.8	0	10	54	0	29.9
11	1	0.166237	72	29	6.8	0	10	64	0	29.92
12	1	0.0657216	72	29	6.8	0	10	23	9	29.88
13	1	0.29768	72	29	6.8	0	10	30	18	29.84
14	1	0.529639	72	29	6.8	0	10	65	11	29.84
15	0	0.761598	72	29	6.8	0	10	75	5	29.85
16	0	0.865459	73	29	7.9	0	10	72	6	29.84
17	0	0.6326	73	29	7.9	0	10	78	6	29.86
18	1	0.399741	73	29	7.9	0	10	63	0	29.88
19	1	0.166882	73	29	7.9	0	10	69	3	29.88
20	1	0.0659767	73	29	7.9	0	10	33	10	29.84
21	1	0.298836	73	29	7.9	0	10	27	20	29.81
22	1	0.531695	73	29	7.9	0	10	53	14	29.81
23	0	0.764554	73	29	7.9	0	10	70	6	29.85
24	0	0.867704	76	30	6.9	0	10	73	0	29.83

dependent variable

Index	Power Generated
0	0
1	0
2	5418
3	25477
4	30069
5	16280
6	515
7	0
8	0
9	0
10	4939
11	24335
12	29025
13	15408
14	491
15	0
16	0
17	0
18	4854
19	23855
20	28339
21	15308
22	455
23	0
24	0

Datasets Train and Test

x_train - DataFrame

Index	s Daylight	ıce to Solar	Temperat	/ind Direc	Wind Spe	Sky Cover	Visibility	tive Humi
0	0	0.859897	69	28	7.5	0	10	75
2	1	0.397172	69	28	7.5	0	10	70
3	1	0.16581	69	28	7.5	0	10	33
5	1	0.296915	69	28	7.5	0	10	20
7	0	0.75964	69	28	7.5	0	10	49
8	0	0.862113	72	29	6.8	0	10	67

x_test - DataFrame

Index	s Daylight	ıce to Solar	Temperat	/ind Direc	Wind Spe	Sky Cover	Visibility	tive Humi
1	0	0.628535	69	28	7.5	0	10	77
4	1	0.0655527	69	28	7.5	0	10	21
6	1	0.528278	69	28	7.5	0	10	36
9	0	0.630155	72	29	6.8	0	10	49
10	1	0.398196	72	29	6.8	0	10	54
14	1	0.529639	72	29	6.8	0	10	65
15	0	0.761598	72	29	6.8	0	10	75
17	0	0.6326	73	29	7.9	0	10	78
20	1	0.0659767	73	29	7.9	0	10	33
22	1	0.531695	73	29	7.9	0	10	53
23	0	0.764554	73	29	7.9	0	10	70
27	1	0.167315	76	30	6.9	0	10	64
30	1	0.533074	76	30	6.9	0	10	27
33	0	0.635417	77	29	8.5	0	10	70

y_train - DataFrame

Index	ıer Genera
0	0
2	5418
3	25477
5	16280
7	0
8	0

y_test - DataFrame

Index	ıer Genera
1	0
4	30069
6	515
9	0
10	4939
14	491
15	0
17	0
20	28339
22	455
23	0
27	23555
30	429
33	0

Codes

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  5 12:55:34 2020

@author: mfa
"""
## 1//Kütüphaneler <<
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

veriler = pd.read_csv("pvdata.csv")
veri1 = veriler.iloc[:, 6:13]
veri2 = veriler.iloc[:, 15:]


dl = veriler.iloc[:,5:6].values
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
dl[:,0] = le.fit_transform(veriler.iloc[:,5:6])
ohe = preprocessing.OneHotEncoder()
dl = ohe.fit_transform(dl).toarray()
dllast = pd.DataFrame(data=dl[:,1:], index=range(2920), columns=['Is Daylight'])
veri1son = pd.concat([dllast,veri1], axis=1)

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(veri1son, veri2, test_size=0.33, random_state=0)

x_train = x_train.sort_index()
x_test = x_test.sort_index()
y_train = y_train.sort_index()
y_test = y_test.sort_index()

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(x_train)
X_test = sc.fit_transform(x_test)
Y_train = sc.fit_transform(y_train)
Y_test = sc.fit_transform(y_test)

from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X_train, Y_train)
y_tahmin = lin_reg.predict(X_test)

is_dllast_list = list(x_test["Is Daylight"])
for counter in range(len(x_test)):
    if is_dllast_list[counter] == 0.0:
        y_tahmin[counter] = 0.0

plt.plot(y_tahmin[0:20,:], color = 'blue')
plt.plot(Y_test[0:20,:], color = 'red')
plt.title("PV Generation Forecasting Linear Regression")
plt.show()

from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
print(r2_score(Y_test, y_tahmin),"\n")
print(mean_squared_error(Y_test, y_tahmin, squared=False),"\n")
print(mean_absolute_error(Y_test, y_tahmin),"\n")

import statsmodels.api as sm
X = np.append(arr = np.ones((2920,1)).astype(int), values=veri1son, axis=1)
X_l = veri1son.iloc[:, [0,1,2,3,4,5,6,7]].values
X_l = np.array(X_l,dtype=float)
model = sm.OLS(veri2, X_l).fit()
print(model.summary(), '\n')
```

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  5 12:55:34 2020

@author: mfa
"""
## 1//Kütüphaneler <<
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


veriler = pd.read_csv("pvdata.csv")
veri1 = veriler.iloc[:, 6:13]
veri2 = veriler.iloc[:, 15:]


dl = veriler.iloc[:,5:6].values
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
dl[:,0] = le.fit_transform(veriler.iloc[:,5:6])
ohe = preprocessing.OneHotEncoder()
dl = ohe.fit_transform(dl).toarray()
dllast = pd.DataFrame(data=dl[:,1:], index=range(2920), columns=['Is Daylight'])
veri1son = pd.concat([dllast,veri1], axis=1)


from sklearn.preprocessing import StandardScaler
sc1 = StandardScaler()
x_olcek = sc1.fit_transform(veri1son)
sc2 = StandardScaler()
y_olcek = sc2.fit_transform(veri2)


from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree=2)
x_poly2 = poly_reg.fit_transform(x_olcek)
lin_reg2 = LinearRegression()
lin_reg2.fit(x_poly2, y_olcek)
y_tahmin2 = lin_reg2.predict(x_poly2)

from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
print("2.derece polinom tahmin:", r2_score(y_olcek, y_tahmin2),"\n")

plt.plot(y_tahmin2[0:20,:], color = 'blue')
plt.plot(y_olcek[0:20,:], color = 'red')
plt.show()

"""4.derece polinom  <<< """
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree=4)
x_poly4 = poly_reg.fit_transform(x_olcek)
lin_reg4 = LinearRegression()
lin_reg4.fit(x_poly4, y_olcek)
y_tahmin4 = lin_reg4.predict(x_poly4)

from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
print("4.derece polinom tahmin:", r2_score(y_olcek, y_tahmin4),"\n")
```

Regression Methods and Results

- Multi Linear Regression
- Support Vector Regression
- Polinomial Regression
  - 2.Degree
  - 4.Degree

```
PV Generation Forecasting with Linear Regression Method: 0.6619942662010241
```

```
PV Generation Forecasting with Linear Regression Method: 0.7457413330617766
```

```
PV Generation Forecasting with 2.degree Polinomial Regression Method: 0.868841677991917
```

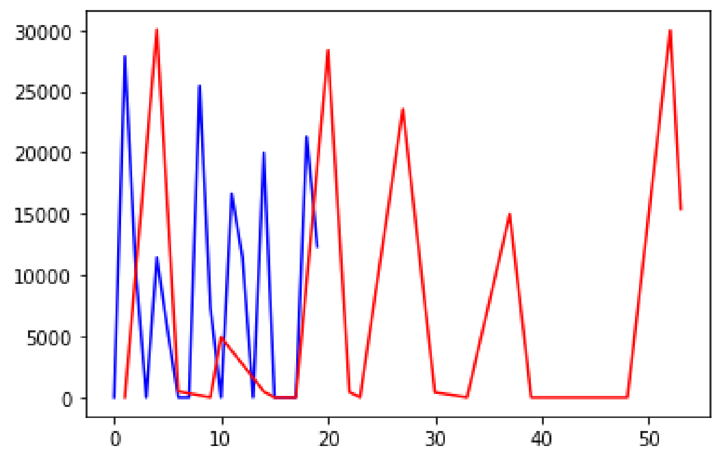
```
PV Generation Forecasting with Support Vector Regression Method: 0.9021700251450911
```

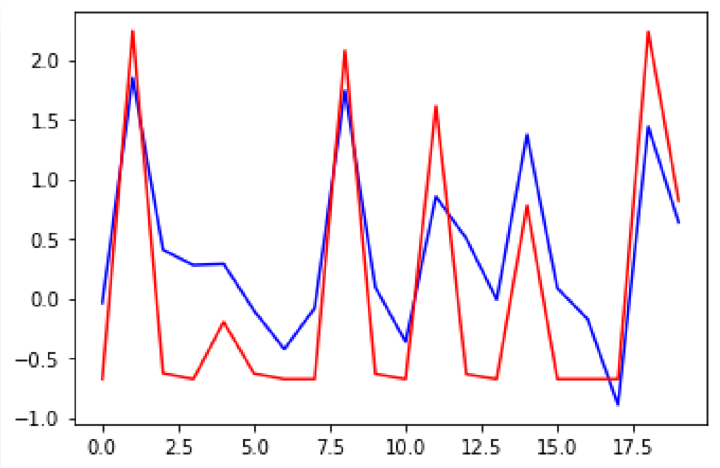
```
PV Generation Forecasting with 4.degree Polinomial Regression Method: 0.915734948862559
```

```
PV Generation Forecasting with 4.degree Polinomial Regression Method: 0.918446515722637
```

Graphs

1





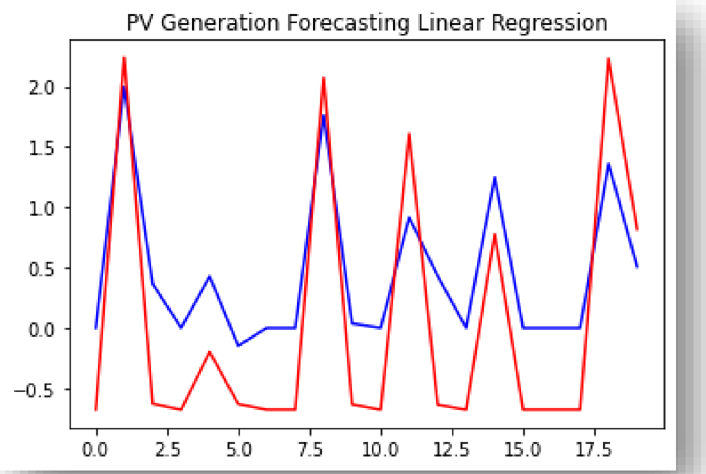
2

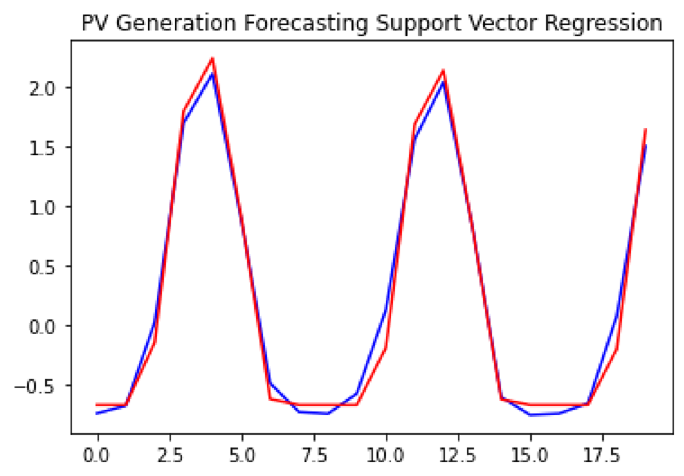
PV Generation

Forecasting

Xlabel:
>Plot Number
Ylabel:
>PV Generation

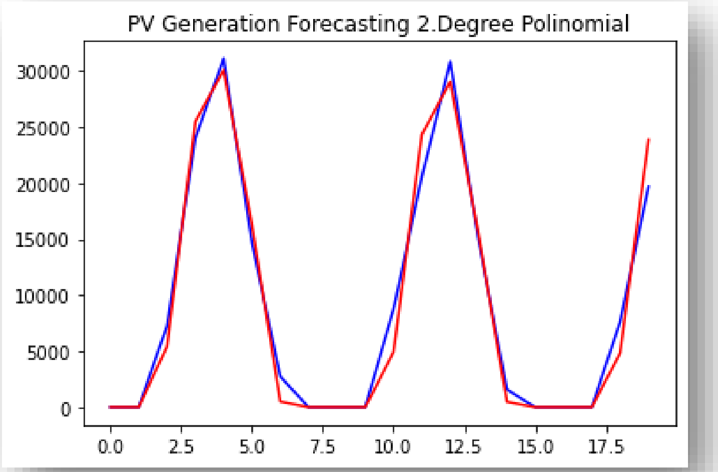
3

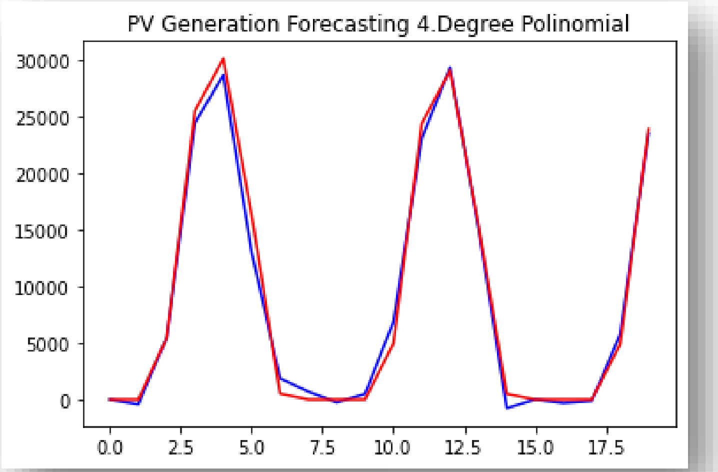


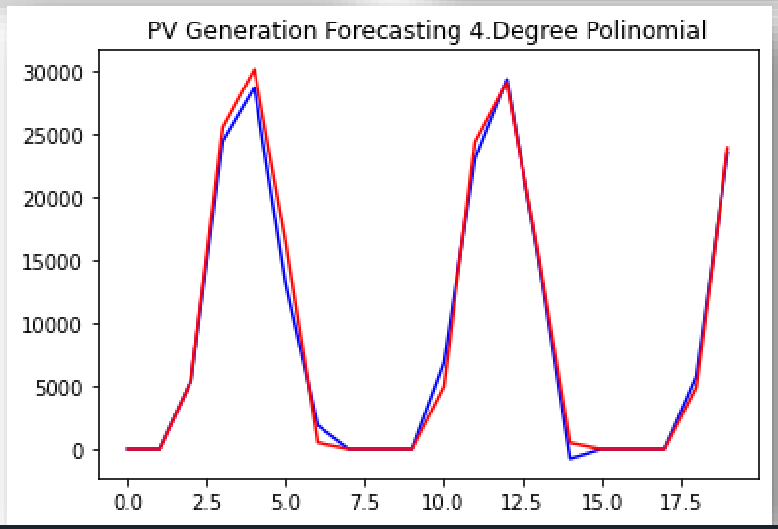


4

Final Graphs








```
PV Generation Forecasting with 4.degree Polinomial Regression Method: 0.918446515722637
```

PV Generation

Forecasting

Xlabel:
>Plot Number
Ylabel:
>Forecasting

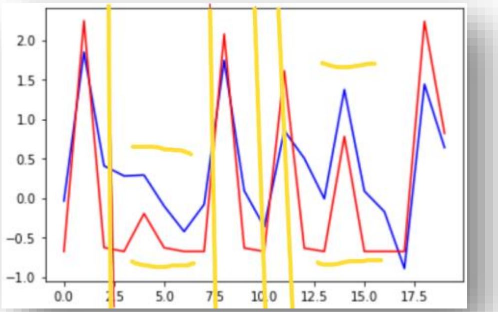
What was planned what was achieved?

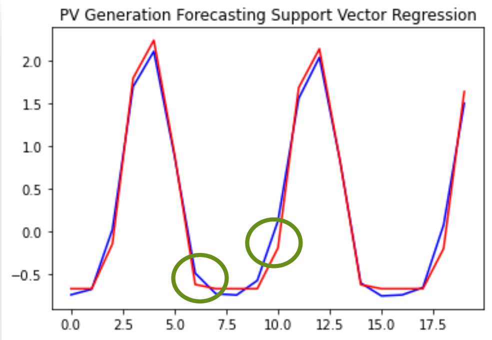
- ## R-squared

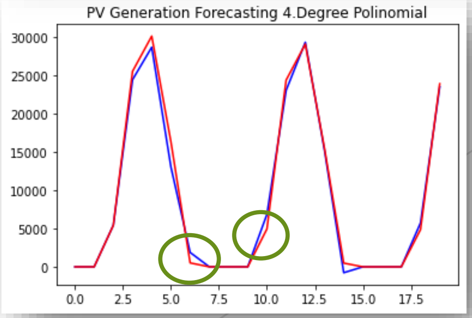
$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$
$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}.$$

- ## OLS Report

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Power Generated   R-squared (uncentered):          0.779
Model:                            OLS   Adj. R-squared (uncentered):     0.779
Method:                 Least Squares   F-statistic:                     1028.
Date:                Tue, 26 Jan 2021   Prob (F-statistic):               0.00
Time:                        19:32:42   Log-Likelihood:                -29471.
No. Observations:                2920   AIC:                         5.896e+04
Df Residuals:                    2910   BIC:                         5.902e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1          6601.2157    405.306     16.287      0.000    5806.500    7395.932
x2         -3.142e+04    677.576    -46.369      0.000   -3.27e+04   -3.01e+04
x3           -95.3294     17.283     -5.516      0.000    -129.217     -61.442
x4            43.0348     17.888      2.406      0.016       7.960      78.110
x5           -90.9553     33.222     -2.738      0.006    -156.095     -25.815
x6          -786.7503     88.011     -8.939      0.000    -959.320    -614.181
x7           119.0875     87.565      1.360      0.174     -52.608     290.783
x8          -157.6229      9.362    -16.836      0.000    -175.980    -139.266
x9           178.2200     21.461      8.305      0.000     136.141     220.299
x10         1195.8944     49.231     24.292      0.000    1099.364    1292.425
==============================================================================
Omnibus:                       13.587   Durbin-Watson:                   1.148
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               16.347
Skew:                           0.085   Prob(JB):                     0.000282
Kurtosis:                       3.325   Cond. No.                         725.
==============================================================================
```



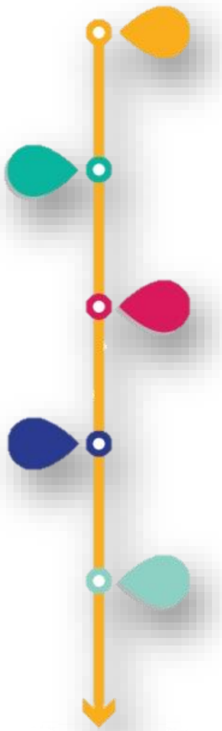




Current Progress and Next Semester Plans

TIMELINE (Months)

Work package name	Fall Term					Spring Term				
	9	10	11	12	01	02	03	04	05	06
Research on the project	X	X								
Review of Python tutorials and books for Machine Learning			X							
Developing Prediction Algorithms			X	X	X					
Mid-semester Presentation			X							
End-of-Semester Report and Presentation										
Arrangement of a small solar panel circuit with Camera and sensor network						X				
Data collection and development of predictions						X	X	X		
Developing Prediction Algorithms								X	X	
Final Report and Presentation										X



Final Results

0.6619942662010241

0.918446515722637

Thank you for Listening