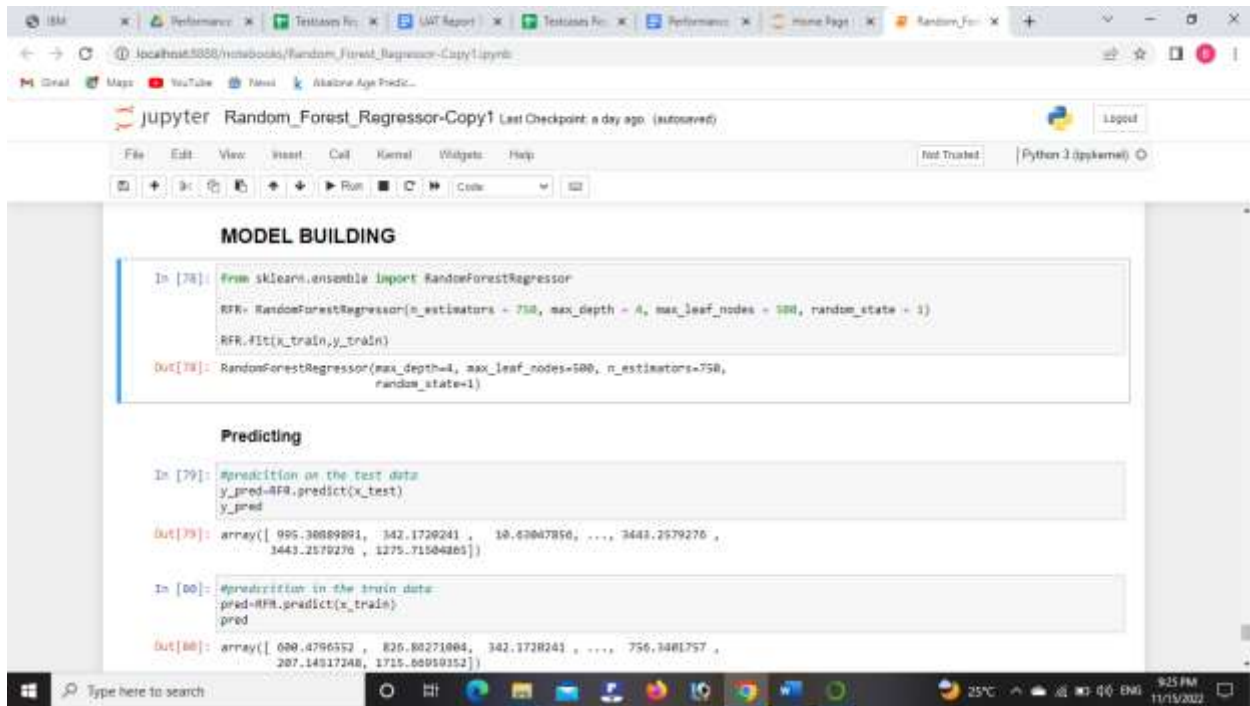


## Project Development Phase Performance Test

Date	15 November 2022
Team ID	PNT2022TMID45553
Project Name	Predicting the energy output of wind turbine based on weather condition
Maximum Marks	10 Marks

### Performance Testing: Regression Model:



The screenshot displays a Jupyter Notebook interface with the following content:

#### MODEL BUILDING

```
In [78]: from sklearn.ensemble import RandomForestRegressor
RFR= RandomForestRegressor(n_estimators = 750, max_depth = 4, max_leaf_nodes = 500, random_state = 1)
RFR.fit(x_train,y_train)
Out[78]: RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                                random_state=1)
```

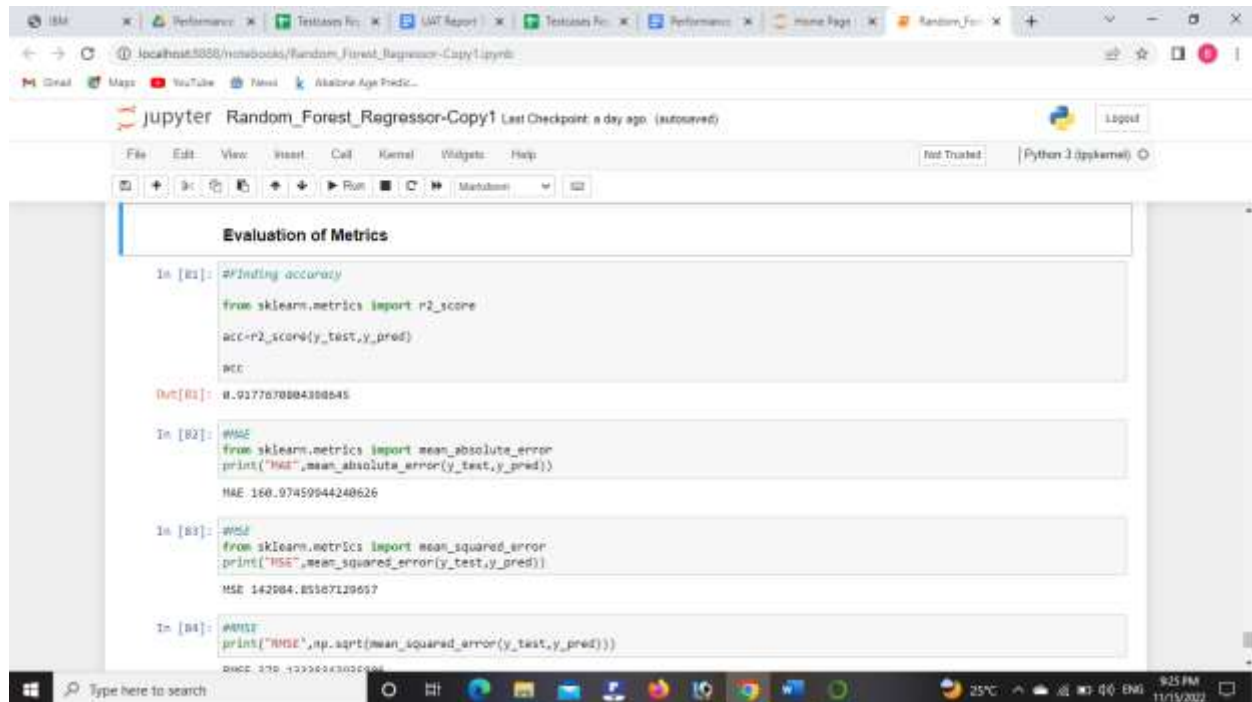
#### Predicting

```
In [79]: #prediction on the test data
y_pred=RFR.predict(x_test)
y_pred
Out[79]: array([ 995.30809091, 342.1720241 , 10.63047850, ..., 3443.2579270 ,
                  3443.2579270 , 1275.71504805])
```

```
In [80]: #prediction in the train data
pred=RFR.predict(x_train)
pred
Out[80]: array([ 600.4796552 , 826.80271004, 342.1720241 , ..., 756.3481757 ,
                  207.14017248, 1715.60910152])
```

## Evaluation Metrics:

MAE - , MSE - , RMSE - , R2 score , Accuracy Score.



A screenshot of a Jupyter Notebook titled "Random\_Forest\_Regressor-Copy1" running on a local host. The notebook contains four code cells for evaluating model performance. The first cell calculates the R2 score using `r2_score` from `sklearn.metrics`, resulting in 0.9177670884398845. The second cell calculates the Mean Absolute Error (MAE) using `mean_absolute_error`, resulting in 160.97459544248626. The third cell calculates the Mean Squared Error (MSE) using `mean_squared_error`, resulting in 142984.85167129657. The fourth cell calculates the Root Mean Squared Error (RMSE) using `np.sqrt(mean_squared_error)`, resulting in 378.13338543935886. The notebook interface shows a file explorer, menu bar, and toolbar at the top, and a Windows taskbar at the bottom.

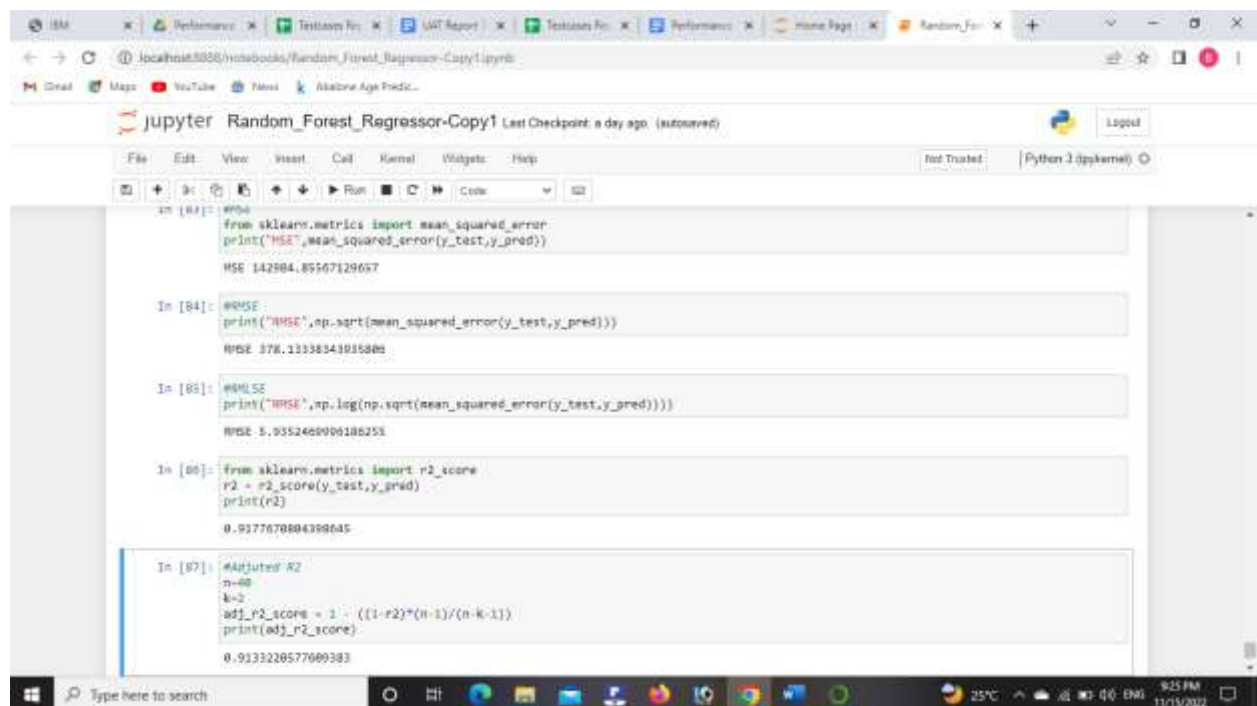
```
#Finding accuracy
from sklearn.metrics import r2_score
acc=r2_score(y_test,y_pred)

R2: 0.9177670884398845

#MAE
from sklearn.metrics import mean_absolute_error
print("MAE",mean_absolute_error(y_test,y_pred))
MAE 160.97459544248626

#MSE
from sklearn.metrics import mean_squared_error
print("MSE",mean_squared_error(y_test,y_pred))
MSE 142984.85167129657

#RMSE
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
RMSE 378.13338543935886
```



A screenshot of a Jupyter Notebook titled "Random\_Forest\_Regressor-Copy1" running on a local host. The notebook contains four code cells for evaluating model performance. The first cell calculates the Mean Squared Error (MSE) using `mean_squared_error` from `sklearn.metrics`, resulting in 142984.85167129657. The second cell calculates the Root Mean Squared Error (RMSE) using `np.sqrt(mean_squared_error)`, resulting in 378.13338543935886. The third cell calculates the Adjusted R2 score using `np.log(np.sqrt(mean_squared_error))`, resulting in 0.9352469906185253. The fourth cell calculates the R2 score using `r2_score` from `sklearn.metrics`, resulting in 0.9177670884398845. The notebook interface shows a file explorer, menu bar, and toolbar at the top, and a Windows taskbar at the bottom.

```
#MSE
from sklearn.metrics import mean_squared_error
print("MSE",mean_squared_error(y_test,y_pred))
MSE 142984.85167129657

#RMSE
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
RMSE 378.13338543935886

#Adjusted R2
print("Adjusted R2",np.log(np.sqrt(mean_squared_error(y_test,y_pred))))
Adjusted R2 0.9352469906185253

#R2
from sklearn.metrics import r2_score
r2 = r2_score(y_test,y_pred)
print(r2)
R2 0.9177670884398845

#Adjusted R2
n=40
k=2
adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))
print(adj_r2_score)
Adjusted R2 0.9133228577069383
```