importing libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

dataset importing

In [2]:

```
df=pd.read_csv('Combined_Cycle_Power_Plant.csv')
X=df.iloc[:,:-1].values
Y=df.iloc[:,-1].values
```

Splitting the dataset into the Training & Test set

In [5]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0
```

Polynomial Regression model Training on the Training set

In [8]:

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=6)
x_poly=poly_reg.fit_transform(x_train,y_train)
lr=LinearRegression()
lr.fit(x_poly,y_train)
```

Out[8]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Predicting the test results

In [26]:

```
y_pred = lr.predict(poly_reg.transform(X_test))
np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[434.76 431.23]
 [457.99 460.01]
 [465.89 461.14]
 ...
 [469.87 473.26]
 [438.61 438.  ]
 [463.19 463.28]]
```

checking the efficiency of the model

In [30]:

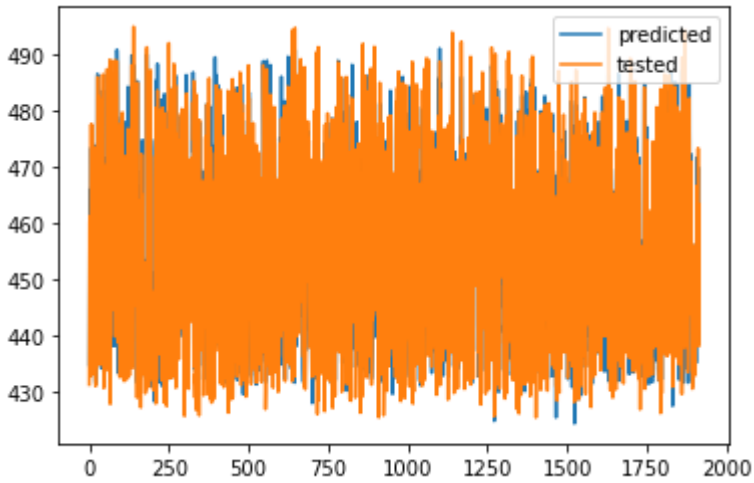
```
from sklearn.metrics import r2_score,mean_absolute_error as mae, mean_squared_error as mse
print(r2_score(y_pred,y_test))
print(mae(y_pred,y_test))
print(np.sqrt(mse(y_pred,y_test)))
```

```
0.9430691622034412
3.106988770053839
3.9759924114631895
```

visualization

In [35]:

```
plt.plot(y_pred)
plt.plot(y_test)
plt.legend(['predicted','tested'])
plt.show()
```



In [ ]:

```

```