In [43]:
```
mean_Thermal = mean_power[['Thermal Generation Estimated (in MU)','Thermal Generation Actual (in MU)']]
mean_Thermal
```

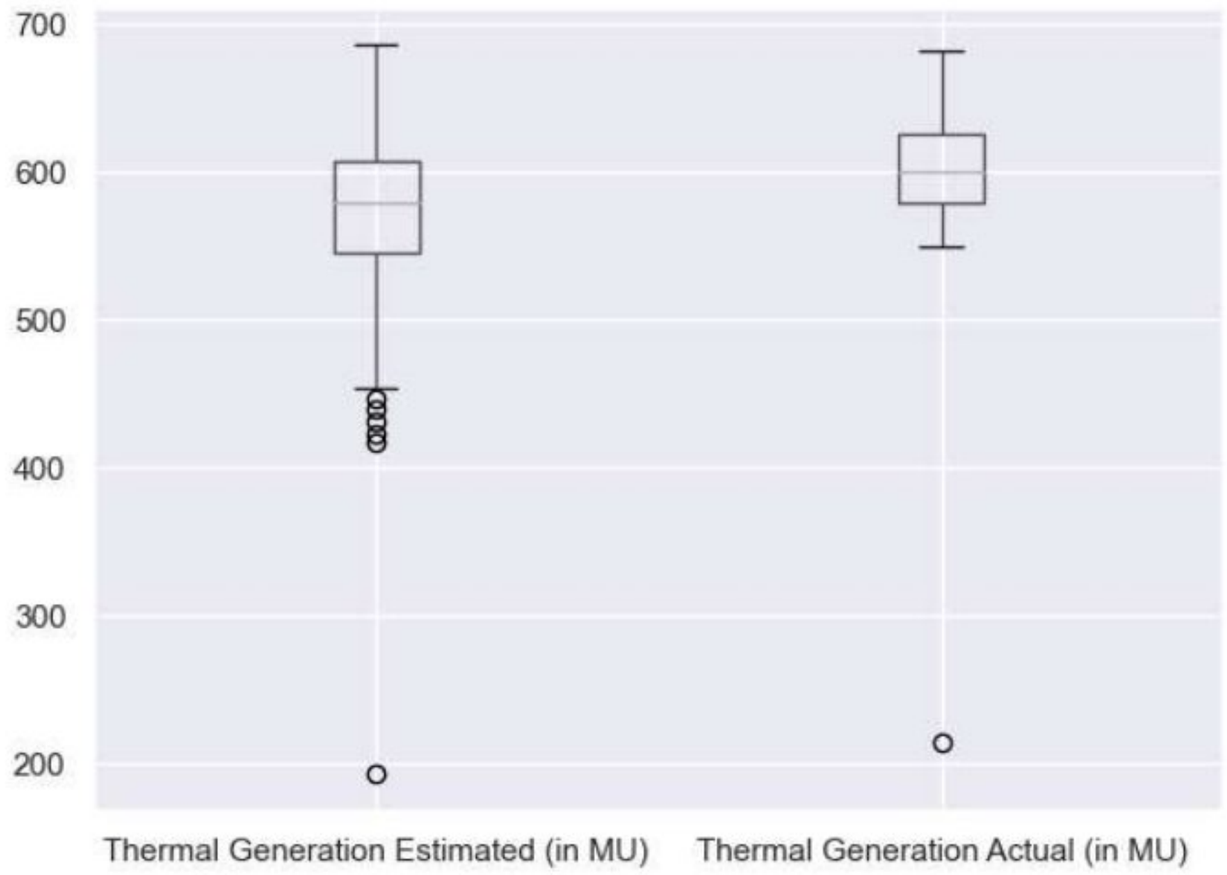
Out[43]:

	Thermal Generation Estimated (in MU)	Thermal Generation Actual (in MU)
0	506.478	555.582
1	512.674	555.582
2	506.646	555.554
3	542.856	555.554
4	555.930	558.170
...	...	...
984	597.208	592.326
985	583.772	592.326
986	576.776	594.872
987	588.622	596.800
988	572.086	559.482

989 rows × 2 columns

In [44]:
```
mean_Thermal.boxplot()
```

Out[44]: <AxesSubplot:>



Creating the training and testing data for linear regression

In [45]:
```
x = mean_Thermal[['Thermal Generation Estimated (in MU)']].values
y = mean_Thermal[['Thermal Generation Actual (in MU)']].values

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

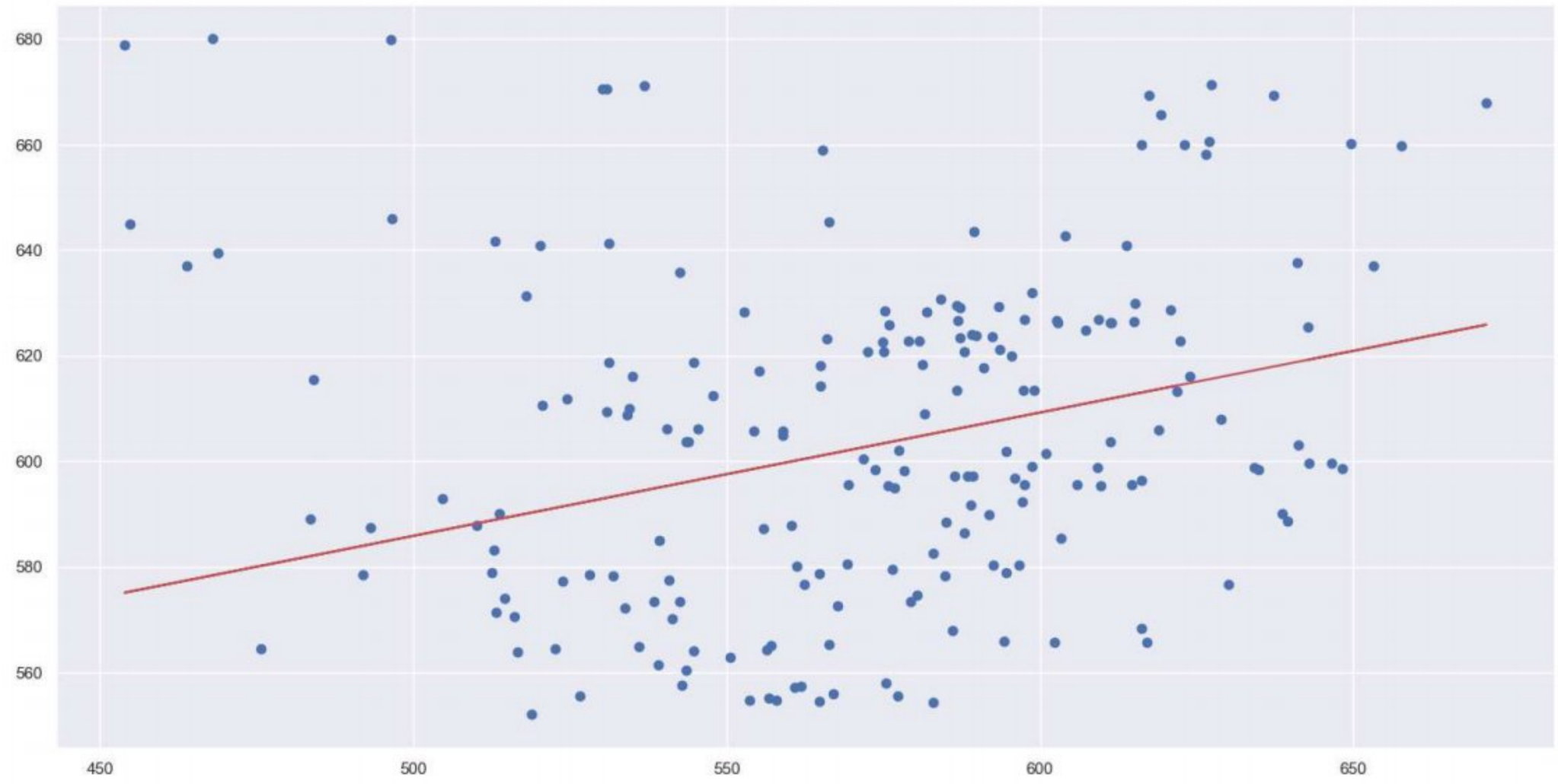
In [46]:
```
linreg = LinearRegression()
linreg.fit(x_train,y_train)
```

Out[46]: LinearRegression()

In [47]:
```
y_pred = linreg.predict(x_test)
```

In [48]:
```
plt.figure(figsize=(18,9))

plt.scatter(x_test,y_test)
plt.plot(x_test,y_pred,color='r')
plt.show()
```



In [49]:
```
linreg.score(x_train,y_train)
```

Out[49]: 0.10360136710745893

In [50]:
```
from sklearn.metrics import r2_score

score = r2_score(y_test,y_pred)

print(f'Test score : {score}')
```

```
Test score : -0.010328087696342969
```

In [51]:
```
f'Slope : {linreg.coef_}'
```

Out[51]: 'Slope : [[0.2336194]]'

In [52]:
```
f'Intercept : {linreg.intercept_}'
```

Out[52]: 'Intercept : [469.043631]'

In [53]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 25.349715449727356
MSE: 984.9802226617126
RMSE: 31.384394572170937
```

In [ ]: