Adrián Landaverde Nava

Cristian Gonzaga López

Andrés Morones

Joel Erick Martínez Espinosa

00. Introduction

In the R Shiny Dashboard made previously: (https://adrian-landaverde.shinyapps.io/CrystalCaseStudy/) we made the EDA. In which we could see how the energy from 2005 and 2006 is very different despite the variables are very alike.

So it will be attempted to predict the energy using 2 approaches:

In the fisrt one, it will be implemented one model for each year using the humidity and the temperature

In the second one, it will be used a time series analysis to predict the energy based on the trend and seasonality of the data

01. Libraries and Data

In [ ]:
```
!pip install statsmodels
```

```
Requirement already satisfied: statsmodels in /usr/local/lib/python3.7/dist-pa
ckages (0.10.2)
Requirement already satisfied: scipy>=0.18 in /usr/local/lib/python3.7/dist-pa
ckages (from statsmodels) (1.4.1)
Requirement already satisfied: patsy>=0.4.0 in /usr/local/lib/python3.7/dist-p
ackages (from statsmodels) (0.5.2)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-pa
ckages (from statsmodels) (1.21.5)
Requirement already satisfied: pandas>=0.19 in /usr/local/lib/python3.7/dist-p
ackages (from statsmodels) (1.3.5)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-p
ackages (from pandas>=0.19->statsmodels) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python
3.7/dist-packages (from pandas>=0.19->statsmodels) (2.8.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from patsy>=0.4.0->statsmodels) (1.15.0)
```

In [2]:
```
import pandas as pd
import numpy as np
```

```
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.pyplot import figure
import seaborn as sns
from scipy import stats
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import MinMaxScaler
import statistics
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: Futur
eWarning: pandas.util.testing is deprecated. Use the functions in the public A
PI at pandas.testing instead.
  import pandas.util.testing as tm
```

In [ ]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [6]:

```
df_energy= pd.read_csv("/content/drive/MyDrive/Universidad/Cuarto Semestre/Cry
df_energy
```

Out[6]:

	Hour	energy_consumpt_2005	energy_consumpt_2006	full_temp_2005	full_humid_2005
0	1	631.623161	1246.300847	-0.400000	64.00000(
1	2	534.397104	1062.500558	-0.733333	65.33333
2	3	453.538785	884.586887	-1.066667	66.66666
3	4	400.699718	786.564121	-1.400000	68.00000(
4	5	378.171092	742.669614	-1.666667	60.33333
...	...	...	...	...	..
8779	8780	950.369306	0.000000	3.333333	64.00000(
8780	8781	880.138770	0.000000	2.666667	68.00000(
8781	8782	792.754026	0.000000	2.000000	72.00000(
8782	8783	740.446668	0.000000	1.333333	76.00000(
8783	8784	706.176769	0.000000	0.666667	80.00000(

8784 rows × 7 columns

In [ ]:
```
df_energy["Year"]=2005
df_energy["Day"]= (((df_energy["Hour"]-1)/24)+1).apply(np.floor)
df_energy["Week"]= (((df_energy["Day"]-1)/7)+1).apply(np.floor)
df_energy
```

Out[ ]:

	Hour	energy_consumpt_2005	energy_consumpt_2006	full_temp_2005	full_humid_2005
0	1	631.623161	1246.300847	-0.400000	64.000000
1	2	534.397104	1062.500558	-0.733333	65.333333
2	3	453.538785	884.586887	-1.066667	66.666667
3	4	400.699718	786.564121	-1.400000	68.000000
4	5	378.171092	742.669614	-1.666667	60.333333
...	...	...	...	...	..
8779	8780	950.369306	0.000000	3.333333	64.000000
8780	8781	880.138770	0.000000	2.666667	68.000000
8781	8782	792.754026	0.000000	2.000000	72.000000
8782	8783	740.446668	0.000000	1.333333	76.000000
8783	8784	706.176769	0.000000	0.666667	80.000000

8784 rows × 10 columns

In [ ]:
```
df_energy["Date"]=(np.asarray(df_energy['Year'], dtype='datetime64[Y]')-1970)
df_energy["Month"]=pd.DatetimeIndex(df_energy['Date']).month
df_energy
```

Out[ ]:

	Hour	energy_consumpt_2005	energy_consumpt_2006	full_temp_2005	full_humid_2005
0	1	631.623161	1246.300847	-0.400000	64.000000
1	2	534.397104	1062.500558	-0.733333	65.333333
2	3	453.538785	884.586887	-1.066667	66.666667
3	4	400.699718	786.564121	-1.400000	68.000000
4	5	378.171092	742.669614	-1.666667	60.333333
...	...	...	...	...	..
8779	8780	950.369306	0.000000	3.333333	64.000000

8780	8781	880.138770	0.000000	2.666667	68.00000(
8781	8782	792.754026	0.000000	2.000000	72.00000(
8782	8783	740.446668	0.000000	1.333333	76.00000(
8783	8784	706.176769	0.000000	0.666667	80.00000(

8784 rows × 12 columns

In [ ]:
```
df_energy.isna().sum()
```

Out[ ]:
```
Hour                   0
energy_consumpt_2005    34
energy_consumpt_2006    42
full_temp_2005          0
full_humid_2005         0
full_temp_2006          24
full_humid_2006         24
Year                   0
Day                    0
Week                   0
Date                   0
Month                  0
dtype: int64
```

In [ ]:
```
df_energy["energy_consumpt_2005"].fillna(value=df_energy["energy_consumpt_200
df_energy["energy_consumpt_2006"].fillna(value=df_energy["energy_consumpt_200
df_energy=df_energy.iloc[:-24]
df_energy.isna().sum()
```

Out[ ]:
```
Hour                   0
energy_consumpt_2005    0
energy_consumpt_2006    0
full_temp_2005          0
full_humid_2005         0
full_temp_2006          0
full_humid_2006         0
Year                   0
Day                    0
Week                   0
Date                   0
Month                  0
dtype: int64
```

In [ ]:
```
variables=["energy_consumpt_2005","energy_consumpt_2006","full_temp_2005","fu
```

02. Machine Learning Regression

Based on the dashboard plots, the energy is more correlated to the mean temperature

per day, and to the humidity per hour

In [ ]:
```
meanEnergy2005= df_energy.groupby("Day")['full_temp_2005'].mean()
meanEnergy2006= df_energy.groupby("Day")['full_temp_2006'].mean()
df_energy["temp_per_day_2005"]=(np.repeat(meanEnergy2005,24)).values
df_energy["temp_per_day_2006"]=(np.repeat(meanEnergy2006,24)).values
df_energy["HourOfDay"]=df_energy["Hour"]-((df_energy['Day']-1)*24)
df_energy
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports un
til
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

Out[ ]:

	Hour	energy_consumpt_2005	energy_consumpt_2006	full_temp_2005	full_humid_2005
0	1	631.623161	1246.300847	-0.400000	64.000000
1	2	534.397104	1062.500558	-0.733333	65.333333
2	3	453.538785	884.586887	-1.066667	66.666667
3	4	400.699718	786.564121	-1.400000	68.000000
4	5	378.171092	742.669614	-1.666667	60.333333
...	...	...	...	...	..
8755	8756	927.986980	1933.241699	1.466667	43.333333
8756	8757	909.729665	1763.838385	0.933333	46.666667
8757	8758	848.714914	1569.071986	0.400000	50.000000

8758	8759	740.913139	1468.273109	-0.066667	54.00000(
8759	8760	613.226705	1384.624630	-0.533333	58.00000(

8760 rows × 15 columns

In [ ]:
```
def plotRgresion(y,y_predict,ax,limites,ymin,ymax):
  ax.plot(df_energy["Hour"],y,label="y Real",color="blue")
  ax.plot(df_energy["Hour"],y_predict,label="y Predicted",color="red")
  if(limites):
    ax.set_ylim(ymin,ymax)
  return(ax)
```

In [ ]:
```
X_2005= df_energy[['temp_per_day_2005','HourOfDay','full_humid_2005']]
y_2005= df_energy['energy_consumpt_2005']

X_train, X_test, y_train, y_test= train_test_split(X_2005,y_2005,random_state
```

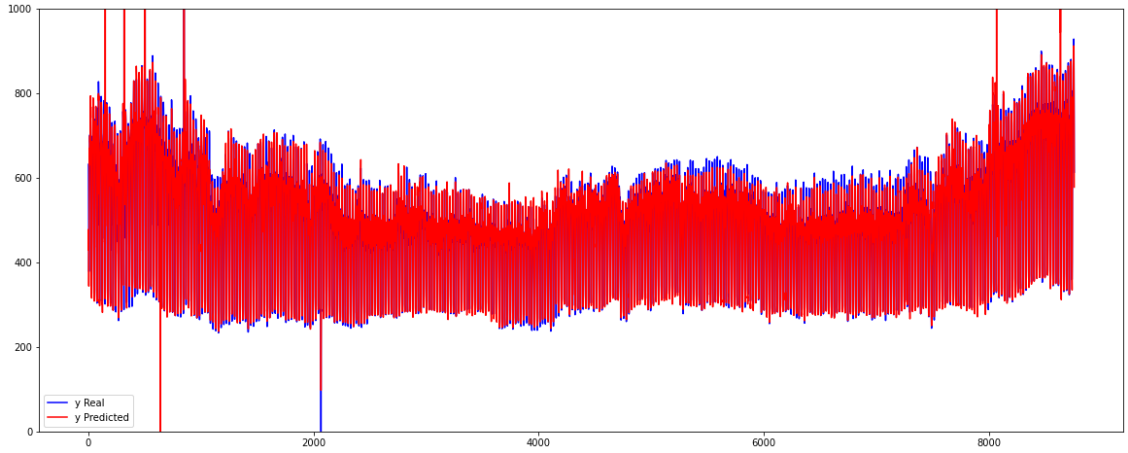
02.1 Random Forest

In [ ]:
```
parameters= {"n_estimators":list(range(10,101,10))}
model = RandomForestRegressor(random_state=0)

grid = GridSearchCV(model, param_grid = parameters,scoring="r2",verbose=3)
grid.fit(X_train,y_train)
print('Grid best parameter (max. R2): ', grid.best_params_)
print('Grid best score (R2): ', grid.best_score_)
print("R2 Score on test:",r2_score(y_test, grid.predict(X_test)))
fig, ax= plt.subplots(figsize=(20,8))
ax= plotRgresion(y_2005,grid.predict(X_2005),ax,True,0,1000)
plt.legend()
plt.show()
print("Final R2 Score:",r2_score(y_2005,grid.predict(X_2005)))
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV 1/5] END ..................n_estimators=10;, score=0.662 total time=   0.
1s
[CV 2/5] END ..................n_estimators=10;, score=0.235 total time=   0.
1s
[CV 3/5] END ..................n_estimators=10;, score=0.924 total time=   0.
1s
[CV 4/5] END ..................n_estimators=10;, score=0.377 total time=   0.
3s
[CV 5/5] END ..................n_estimators=10;, score=0.118 total time=   0.
2s
[CV 1/5] END ..................n_estimators=20;, score=0.670 total time=   0.
5s
[CV 2/5] END ..................n_estimators=20;, score=0.651 total time=   0.
5s
[CV 3/5] END ..................n_estimators=20;, score=0.896 total time=   0.
4s
```

```
[CV 4/5] END ..................n_estimators=20;, score=0.365 total time=   0.
5s
[CV 5/5] END ..................n_estimators=20;, score=0.235 total time=   0.
6s
[CV 1/5] END ..................n_estimators=30;, score=0.670 total time=   0.
7s
[CV 2/5] END ..................n_estimators=30;, score=0.625 total time=   0.
8s
[CV 3/5] END ..................n_estimators=30;, score=0.915 total time=   0.
8s
[CV 4/5] END ..................n_estimators=30;, score=0.371 total time=   0.
6s
[CV 5/5] END ..................n_estimators=30;, score=0.202 total time=   0.
6s
[CV 1/5] END ..................n_estimators=40;, score=0.672 total time=   0.
9s
[CV 2/5] END ..................n_estimators=40;, score=0.701 total time=   0.
8s
[CV 3/5] END ..................n_estimators=40;, score=0.916 total time=   0.
8s
[CV 4/5] END ..................n_estimators=40;, score=0.379 total time=   0.
8s
[CV 5/5] END ..................n_estimators=40;, score=0.266 total time=   1.
0s
[CV 1/5] END ..................n_estimators=50;, score=0.672 total time=   1.
2s
[CV 2/5] END ..................n_estimators=50;, score=0.705 total time=   1.
7s
[CV 3/5] END ..................n_estimators=50;, score=0.924 total time=   1.
6s
[CV 4/5] END ..................n_estimators=50;, score=0.380 total time=   1.
2s
[CV 5/5] END ..................n_estimators=50;, score=0.266 total time=   1.
3s
[CV 1/5] END ..................n_estimators=60;, score=0.672 total time=   1.
8s
[CV 2/5] END ..................n_estimators=60;, score=0.697 total time=   1.
4s
[CV 3/5] END ..................n_estimators=60;, score=0.930 total time=   1.
4s
[CV 4/5] END ..................n_estimators=60;, score=0.377 total time=   1.
3s
[CV 5/5] END ..................n_estimators=60;, score=0.276 total time=   1.
3s
[CV 1/5] END ..................n_estimators=70;, score=0.672 total time=   1.
6s
[CV 2/5] END ..................n_estimators=70;, score=0.736 total time=   0.
9s
[CV 3/5] END ..................n_estimators=70;, score=0.933 total time=   0.
8s
[CV 4/5] END ..................n_estimators=70;, score=0.382 total time=   0.
8s
[CV 5/5] END ..................n_estimators=70;, score=0.275 total time=   0.
8s
[CV 1/5] END ..................n_estimators=80;, score=0.673 total time=   0.
9s
[CV 2/5] END ..................n_estimators=80;, score=0.723 total time=   0.
9s
[CV 3/5] END ..................n_estimators=80;, score=0.931 total time=   0.
9s
[CV 4/5] END                    n_estimators=80;, score=0.385 total time=   0
```

```
[CV 4/5] END ..................n_estimators=80;, score=0.389 total time=   0.
9s
[CV 5/5] END ..................n_estimators=80;, score=0.259 total time=   0.
9s
[CV 1/5] END ..................n_estimators=90;, score=0.673 total time=   1.
0s
[CV 2/5] END ..................n_estimators=90;, score=0.731 total time=   1.
1s
[CV 3/5] END ..................n_estimators=90;, score=0.932 total time=   1.
0s
[CV 4/5] END ..................n_estimators=90;, score=0.388 total time=   1.
0s
[CV 5/5] END ..................n_estimators=90;, score=0.253 total time=   1.
1s
[CV 1/5] END .................n_estimators=100;, score=0.673 total time=   1.
2s
[CV 2/5] END .................n_estimators=100;, score=0.722 total time=   1.
2s
[CV 3/5] END .................n_estimators=100;, score=0.933 total time=   1.
2s
[CV 4/5] END .................n_estimators=100;, score=0.387 total time=   1.
2s
[CV 5/5] END .................n_estimators=100;, score=0.261 total time=   1.
2s
Grid best parameter (max. R2):  {'n_estimators': 70}
Grid best score (R2):  0.5996080433637754
R2 Score on test: 0.6105817867373805
```




```
Final R2 Score: 0.8696535631662909
```

02.2 K-Nearet Neighbours

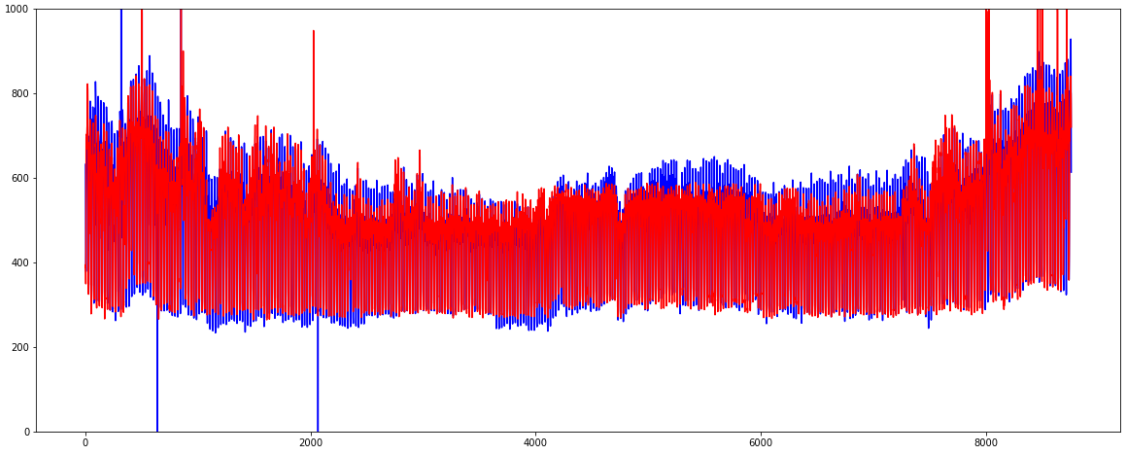
In [ ]:

```
parametrosKNN= {"n_neighbors":list(range(3,19,2))}
knn = KNeighborsRegressor()

grid = GridSearchCV(knn, param_grid = parametrosKNN,scoring="r2",verbose=3)
grid.fit(X_train,y_train)
print('Grid best parameter (max. R2): ', grid.best_params_)
print('Grid best score (R2): ', grid.best_score_)
print("R2 Score on test:",r2_score(y_test, grid.predict(X_test)))
fig, ax= plt.subplots(figsize=(20,8))
ax= plotRgresion(y_2005,grid.predict(X_2005),ax,True,0,1000)
plt.show()
print("Final R2 Score:",r2_score(y_2005,grid.predict(X_2005)))
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
```

```
[CV 1/5] END .....................n_neighbors=3;, score=0.638 total time=   0.
0s
[CV 2/5] END .....................n_neighbors=3;, score=0.515 total time=   0.
0s
[CV 3/5] END .....................n_neighbors=3;, score=0.722 total time=   0.
0s
[CV 4/5] END .....................n_neighbors=3;, score=0.322 total time=   0.
0s
[CV 5/5] END .....................n_neighbors=3;, score=0.382 total time=   0.
0s
[CV 1/5] END .....................n_neighbors=5;, score=0.639 total time=   0.
0s
[CV 2/5] END .....................n_neighbors=5;, score=0.716 total time=   0.
0s
[CV 3/5] END .....................n_neighbors=5;, score=0.821 total time=   0.
0s
[CV 4/5] END .....................n_neighbors=5;, score=0.362 total time=   0.
0s
[CV 5/5] END .....................n_neighbors=5;, score=0.337 total time=   0.
0s
[CV 1/5] END .....................n_neighbors=7;, score=0.632 total time=   0.
0s
[CV 2/5] END .....................n_neighbors=7;, score=0.721 total time=   0.
0s
[CV 3/5] END .....................n_neighbors=7;, score=0.802 total time=   0.
0s
[CV 4/5] END .....................n_neighbors=7;, score=0.370 total time=   0.
0s
[CV 5/5] END .....................n_neighbors=7;, score=0.358 total time=   0.
0s
[CV 1/5] END .....................n_neighbors=9;, score=0.616 total time=   0.
0s
[CV 2/5] END .....................n_neighbors=9;, score=0.773 total time=   0.
0s
[CV 3/5] END .....................n_neighbors=9;, score=0.832 total time=   0.
0s
[CV 4/5] END .....................n_neighbors=9;, score=0.371 total time=   0.
0s
[CV 5/5] END .....................n_neighbors=9;, score=0.360 total time=   0.
0s
[CV 1/5] END ....................n_neighbors=11;, score=0.624 total time=   0.
0s
[CV 2/5] END ....................n_neighbors=11;, score=0.779 total time=   0.
0s
[CV 3/5] END ....................n_neighbors=11;, score=0.849 total time=   0.
0s
[CV 4/5] END ....................n_neighbors=11;, score=0.372 total time=   0.
0s
[CV 5/5] END ....................n_neighbors=11;, score=0.339 total time=   0.
0s
[CV 1/5] END ....................n_neighbors=13;, score=0.627 total time=   0.
0s
[CV 2/5] END ....................n_neighbors=13;, score=0.785 total time=   0.
0s
[CV 3/5] END ....................n_neighbors=13;, score=0.844 total time=   0.
0s
[CV 4/5] END ....................n_neighbors=13;, score=0.374 total time=   0.
0s
[CV 5/5] END ....................n_neighbors=13;, score=0.337 total time=   0.
0s
```

```
[CV 1/5] END ..................n_neighbors=15;, score=0.627 total time=   0.
0s
[CV 2/5] END ..................n_neighbors=15;, score=0.790 total time=   0.
0s
[CV 3/5] END ..................n_neighbors=15;, score=0.850 total time=   0.
0s
[CV 4/5] END ..................n_neighbors=15;, score=0.374 total time=   0.
0s
[CV 5/5] END ..................n_neighbors=15;, score=0.338 total time=   0.
0s
[CV 1/5] END ..................n_neighbors=17;, score=0.625 total time=   0.
0s
[CV 2/5] END ..................n_neighbors=17;, score=0.796 total time=   0.
0s
[CV 3/5] END ..................n_neighbors=17;, score=0.843 total time=   0.
0s
[CV 4/5] END ..................n_neighbors=17;, score=0.371 total time=   0.
0s
[CV 5/5] END ..................n_neighbors=17;, score=0.340 total time=   0.
0s
Grid best parameter (max. R2):  {'n_neighbors': 15}
Grid best score (R2):  0.5958735892803727
R2 Score on test: 0.8486469601628195
```




```
Final R2 Score: 0.6279205764416236
```

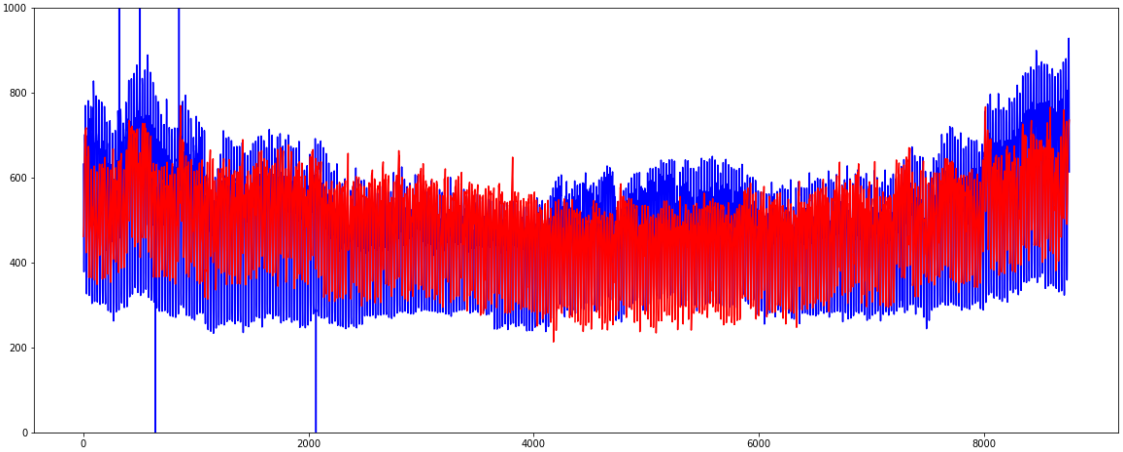
02.3 Linear Regresion

In [ ]:

```
parametrosLR= {"fit_intercept":[True,False],"normalize":[True,False]}
linear = LinearRegression()

grid = GridSearchCV(linear, param_grid = parametrosLR,scoring="r2",verbose=3)
grid.fit(X_train,y_train)
print('Grid best parameter (max. R2): ', grid.best_params_)
print('Grid best score (R2): ', grid.best_score_)
print("R2 Score on test:",r2_score(y_test, grid.predict(X_test)))
fig, ax= plt.subplots(figsize=(20,8))
ax= plotRgresion(y_2005,grid.predict(X_2005),ax,True,0,1000)
plt.show()
print("Final R2 Score:",r2_score(y_2005,grid.predict(X_2005)))
```

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END fit_intercept=True, normalize=True;, score=0.413 total time=   0.
0s
[CV 2/5] END fit_intercept=True, normalize=True;, score=0.540 total time=   0.
0s
```

```
[CV 3/5] END fit_intercept=True, normalize=True;, score=0.583 total time=   0.
0s
[CV 4/5] END fit_intercept=True, normalize=True;, score=0.224 total time=   0.
0s
[CV 5/5] END fit_intercept=True, normalize=True;, score=0.235 total time=   0.
0s
[CV 1/5] END fit_intercept=True, normalize=False;, score=0.413 total time=
0.0s
[CV 2/5] END fit_intercept=True, normalize=False;, score=0.540 total time=
0.0s
[CV 3/5] END fit_intercept=True, normalize=False;, score=0.583 total time=
0.0s
[CV 4/5] END fit_intercept=True, normalize=False;, score=0.224 total time=
0.0s
[CV 5/5] END fit_intercept=True, normalize=False;, score=0.235 total time=
0.0s
[CV 1/5] END fit_intercept=False, normalize=True;, score=-0.037 total time=
0.0s
[CV 2/5] END fit_intercept=False, normalize=True;, score=-0.166 total time=
0.0s
[CV 3/5] END fit_intercept=False, normalize=True;, score=-0.137 total time=
0.0s
[CV 4/5] END fit_intercept=False, normalize=True;, score=-0.135 total time=
0.0s
[CV 5/5] END fit_intercept=False, normalize=True;, score=-0.040 total time=
0.0s
[CV 1/5] END fit_intercept=False, normalize=False;, score=-0.037 total time=
0.0s
[CV 2/5] END fit_intercept=False, normalize=False;, score=-0.166 total time=
0.0s
[CV 3/5] END fit_intercept=False, normalize=False;, score=-0.137 total time=
0.0s
[CV 4/5] END fit_intercept=False, normalize=False;, score=-0.135 total time=
0.0s
[CV 5/5] END fit_intercept=False, normalize=False;, score=-0.040 total time=
0.0s
Grid best parameter (max. R2):  {'fit_intercept': True, 'normalize': True}
Grid best score (R2):  0.3988432306680966
R2 Score on test: 0.5329791947340208
```




```
Final R2 Score: 0.37299626339069825
```

02.4 Neural Networks

In [ ]:

```
layers=[]
```

```
for i in range(1,11):
  for j in range(1,11):
    layers.append((i,j))
parametrosMLP= {"hidden_layer_sizes":layers}
mlp = MLPRegressor(max_iter=5000)

grid = GridSearchCV(mlp, param_grid = parametrosMLP,scoring="r2",verbose=3)
grid.fit(X_train,y_train)
print('Grid best parameter (max. R2): ', grid.best_params_)
print('Grid best score (R2): ', grid.best_score_)
print("R2 Score on test:",r2_score(y_test, grid.predict(X_test)))
fig, ax= plt.subplots(figsize=(20,8))
ax= plotRgresion(y_2005,grid.predict(X_2005),ax,True,0,1000)
plt.show()
print("Final R2 Score:",r2_score(y_2005,grid.predict(X_2005)))
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(1, 1);, score=-5.158 total time=  52.
0s
[CV 2/5] END ........hidden_layer_sizes=(1, 1);, score=-0.000 total time=  42.
0s
[CV 3/5] END ........hidden_layer_sizes=(1, 1);, score=0.582 total time=   8.
7s
[CV 4/5] END ........hidden_layer_sizes=(1, 1);, score=0.221 total time=  12.
9s
[CV 5/5] END ........hidden_layer_sizes=(1, 1);, score=0.234 total time=  22.
0s
[CV 1/5] END ........hidden_layer_sizes=(1, 2);, score=0.414 total time=   9.
4s
[CV 2/5] END ........hidden_layer_sizes=(1, 2);, score=0.540 total time=   9.
4s
[CV 3/5] END ........hidden_layer_sizes=(1, 2);, score=0.583 total time=  10.
2s
[CV 4/5] END ........hidden_layer_sizes=(1, 2);, score=0.223 total time=   7.
2s
[CV 5/5] END ........hidden_layer_sizes=(1, 2);, score=0.235 total time=   9.
8s
[CV 1/5] END ........hidden_layer_sizes=(1, 3);, score=0.410 total time=   6.
5s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 2/5] END ........hidden_layer_sizes=(1, 3);, score=-6.794 total time=  51.
9s
[CV 3/5] END ........hidden_layer_sizes=(1, 3);, score=0.583 total time=   7.
8s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 4/5] END ........hidden_layer_sizes=(1, 3);, score=-2.699 total time=  58.
2s
[CV 5/5] END ........hidden_layer_sizes=(1, 3);, score=0.235 total time=   5.
8s
[CV 1/5] END ........hidden_layer_sizes=(1, 4);, score=0.414 total time=  10.
```

```
6s
[CV 2/5] END .........hidden_layer_sizes=(1, 4);, score=0.540 total time=   7.
1s
[CV 3/5] END .........hidden_layer_sizes=(1, 4);, score=0.581 total time=   7.
8s
[CV 4/5] END .........hidden_layer_sizes=(1, 4);, score=0.221 total time=   5.
8s
[CV 5/5] END .........hidden_layer_sizes=(1, 4);, score=0.235 total time=   9.
0s
[CV 1/5] END .........hidden_layer_sizes=(1, 5);, score=-0.000 total time=   5.
0s
[CV 2/5] END .........hidden_layer_sizes=(1, 5);, score=0.540 total time=  29.
0s
[CV 3/5] END .........hidden_layer_sizes=(1, 5);, score=0.580 total time=   7.
9s
[CV 4/5] END .........hidden_layer_sizes=(1, 5);, score=0.221 total time=   5.
5s
[CV 5/5] END .........hidden_layer_sizes=(1, 5);, score=0.235 total time=   7.
0s
[CV 1/5] END .........hidden_layer_sizes=(1, 6);, score=0.424 total time=  10.
3s
[CV 2/5] END .........hidden_layer_sizes=(1, 6);, score=0.540 total time=   9.
1s
[CV 3/5] END .........hidden_layer_sizes=(1, 6);, score=0.587 total time=  10.
0s
[CV 4/5] END .........hidden_layer_sizes=(1, 6);, score=-0.000 total time=   4.
4s
[CV 5/5] END .........hidden_layer_sizes=(1, 6);, score=0.235 total time=   5.
9s
[CV 1/5] END .........hidden_layer_sizes=(1, 7);, score=0.421 total time=   9.
4s
[CV 2/5] END .........hidden_layer_sizes=(1, 7);, score=0.546 total time=   6.
8s
[CV 3/5] END .........hidden_layer_sizes=(1, 7);, score=0.582 total time=   8.
4s
[CV 4/5] END .........hidden_layer_sizes=(1, 7);, score=0.223 total time=   9.
1s
[CV 5/5] END .........hidden_layer_sizes=(1, 7);, score=0.234 total time=   5.
8s
[CV 1/5] END .........hidden_layer_sizes=(1, 8);, score=0.430 total time=   9.
3s
[CV 2/5] END .........hidden_layer_sizes=(1, 8);, score=0.540 total time=   5.
3s
[CV 3/5] END .........hidden_layer_sizes=(1, 8);, score=0.583 total time=   6.
6s
[CV 4/5] END .........hidden_layer_sizes=(1, 8);, score=-0.000 total time=   3.
7s
[CV 5/5] END .........hidden_layer_sizes=(1, 8);, score=0.235 total time=   7.
0s
[CV 1/5] END .........hidden_layer_sizes=(1, 9);, score=0.412 total time=   6.
1s
[CV 2/5] END .........hidden_layer_sizes=(1, 9);, score=0.540 total time=   7.
2s
[CV 3/5] END .........hidden_layer_sizes=(1, 9);, score=0.581 total time=   6.
5s
[CV 4/5] END .........hidden_layer_sizes=(1, 9);, score=0.222 total time=   7.
8s
[CV 5/5] END .........hidden_layer_sizes=(1, 9);, score=-0.001 total time=   4.
6s
[CV 1/5] END .........hidden_layer_sizes=(1, 10);, score=0.413 total time=   7.
8s
```

```
[CV 2/5] END ........hidden_layer_sizes=(1, 10);, score=0.546 total time=  10.
4s
[CV 3/5] END ........hidden_layer_sizes=(1, 10);, score=0.580 total time=   6.
5s
[CV 4/5] END .......hidden_layer_sizes=(1, 10);, score=-0.000 total time=   4.
8s
[CV 5/5] END .......hidden_layer_sizes=(1, 10);, score=-0.001 total time=   4.
2s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(2, 1);, score=-5.159 total time=  56.
3s
[CV 2/5] END ........hidden_layer_sizes=(2, 1);, score=0.539 total time=   9.
8s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 3/5] END ........hidden_layer_sizes=(2, 1);, score=-6.517 total time=  55.
8s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 4/5] END ........hidden_layer_sizes=(2, 1);, score=-2.723 total time=  56.
2s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 5/5] END ........hidden_layer_sizes=(2, 1);, score=-2.859 total time=  54.
2s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(2, 2);, score=-5.149 total time=  56.
6s
[CV 2/5] END ........hidden_layer_sizes=(2, 2);, score=-0.000 total time=   8.
7s
[CV 3/5] END ........hidden_layer_sizes=(2, 2);, score=0.581 total time=   7.
4s
[CV 4/5] END ........hidden_layer_sizes=(2, 2);, score=0.223 total time=  10.
8s
[CV 5/5] END ........hidden_layer_sizes=(2, 2);, score=0.234 total time=   6.
4s
[CV 1/5] END ........hidden_layer_sizes=(2, 3);, score=0.480 total time=   5.
7s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 2/5] END ........hidden_layer_sizes=(2, 3);, score=-6.754 total time= 1.1m
in
[CV 3/5] END ........hidden_layer_sizes=(2, 3);, score=0.635 total time=   6.
8s
[CV 4/5] END ........hidden_layer_sizes=(2, 3);, score=0.223 total time=  28.
4s
```

```
[CV 5/5] END .........hidden_layer_sizes=(2, 3);, score=0.237 total time=   7.
6s
[CV 1/5] END .........hidden_layer_sizes=(2, 4);, score=0.414 total time=   7.
2s
[CV 2/5] END .........hidden_layer_sizes=(2, 4);, score=0.539 total time=  10.
0s
[CV 3/5] END .........hidden_layer_sizes=(2, 4);, score=0.676 total time=  10.
3s
[CV 4/5] END .........hidden_layer_sizes=(2, 4);, score=0.230 total time=  13.
8s
[CV 5/5] END .........hidden_layer_sizes=(2, 4);, score=0.277 total time=   5.
1s
[CV 1/5] END .........hidden_layer_sizes=(2, 5);, score=0.537 total time=  13.
3s
[CV 2/5] END .........hidden_layer_sizes=(2, 5);, score=0.540 total time=   7.
6s
[CV 3/5] END .........hidden_layer_sizes=(2, 5);, score=0.583 total time=   9.
0s
[CV 4/5] END .........hidden_layer_sizes=(2, 5);, score=0.232 total time=  10.
9s
[CV 5/5] END .........hidden_layer_sizes=(2, 5);, score=0.236 total time=   6.
2s
[CV 1/5] END .........hidden_layer_sizes=(2, 6);, score=0.476 total time=   5.
4s
[CV 2/5] END .........hidden_layer_sizes=(2, 6);, score=0.580 total time=   5.
3s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 3/5] END .........hidden_layer_sizes=(2, 6);, score=-6.489 total time= 1.1m
in
[CV 4/5] END .........hidden_layer_sizes=(2, 6);, score=0.220 total time=  12.
7s
[CV 5/5] END .........hidden_layer_sizes=(2, 6);, score=0.237 total time=   6.
9s
[CV 1/5] END .........hidden_layer_sizes=(2, 7);, score=0.412 total time=   7.
8s
[CV 2/5] END .........hidden_layer_sizes=(2, 7);, score=0.540 total time=   8.
6s
[CV 3/5] END .........hidden_layer_sizes=(2, 7);, score=0.584 total time=   7.
2s
[CV 4/5] END .........hidden_layer_sizes=(2, 7);, score=0.222 total time=   5.
2s
[CV 5/5] END .........hidden_layer_sizes=(2, 7);, score=0.236 total time=   7.
0s
[CV 1/5] END .........hidden_layer_sizes=(2, 8);, score=0.414 total time=   6.
6s
[CV 2/5] END .........hidden_layer_sizes=(2, 8);, score=0.541 total time=   7.
2s
[CV 3/5] END .........hidden_layer_sizes=(2, 8);, score=0.582 total time=   6.
5s
[CV 4/5] END .........hidden_layer_sizes=(2, 8);, score=0.255 total time=   6.
9s
[CV 5/5] END .........hidden_layer_sizes=(2, 8);, score=0.235 total time=   6.
9s
[CV 1/5] END .........hidden_layer_sizes=(2, 9);, score=0.488 total time=   6.
4s
[CV 2/5] END .........hidden_layer_sizes=(2, 9);, score=0.539 total time=   9.
9s
```

```
[CV 3/5] END .........hidden_layer_sizes=(2, 9);, score=0.581 total time=   7.
3s
[CV 4/5] END .........hidden_layer_sizes=(2, 9);, score=0.224 total time=   7.
6s
[CV 5/5] END .........hidden_layer_sizes=(2, 9);, score=0.272 total time=   5.
8s
[CV 1/5] END ........hidden_layer_sizes=(2, 10);, score=0.416 total time=   5.
9s
[CV 2/5] END ........hidden_layer_sizes=(2, 10);, score=0.540 total time=   6.
1s
[CV 3/5] END ........hidden_layer_sizes=(2, 10);, score=0.582 total time=   6.
6s
[CV 4/5] END ........hidden_layer_sizes=(2, 10);, score=0.282 total time=   9.
6s
[CV 5/5] END ........hidden_layer_sizes=(2, 10);, score=0.237 total time=   8.
2s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(3, 1);, score=-5.116 total time=  58.
3s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 2/5] END ........hidden_layer_sizes=(3, 1);, score=-6.727 total time=  59.
8s
[CV 3/5] END .........hidden_layer_sizes=(3, 1);, score=0.582 total time=   8.
5s
[CV 4/5] END .........hidden_layer_sizes=(3, 1);, score=0.223 total time=   9.
9s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 5/5] END ........hidden_layer_sizes=(3, 1);, score=-2.856 total time= 1.0m
in
[CV 1/5] END .........hidden_layer_sizes=(3, 2);, score=0.413 total time=  12.
7s
[CV 2/5] END .........hidden_layer_sizes=(3, 2);, score=0.539 total time=   8.
9s
[CV 3/5] END .........hidden_layer_sizes=(3, 2);, score=0.578 total time=   7.
3s
[CV 4/5] END .........hidden_layer_sizes=(3, 2);, score=0.244 total time=   9.
2s
[CV 5/5] END .........hidden_layer_sizes=(3, 2);, score=0.237 total time=   7.
1s
[CV 1/5] END .........hidden_layer_sizes=(3, 3);, score=0.415 total time=   5.
5s
[CV 2/5] END .........hidden_layer_sizes=(3, 3);, score=0.695 total time=  12.
5s
[CV 3/5] END .........hidden_layer_sizes=(3, 3);, score=0.722 total time=   8.
7s
[CV 4/5] END .........hidden_layer_sizes=(3, 3);, score=0.239 total time=   7.
0s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 5/5] END         hidden_layer_sizes=(3, 3);, score=-2.861 total time= 1.1m
```

```
[CV 5/5] END .....hidden_layer_sizes=(3, 3);, score=-2.861 total time= 1.1m
in
[CV 1/5] END .....hidden_layer_sizes=(3, 4);, score=0.414 total time=   5.
3s
[CV 2/5] END .....hidden_layer_sizes=(3, 4);, score=0.540 total time=   5.
8s
[CV 3/5] END .....hidden_layer_sizes=(3, 4);, score=0.587 total time=   6.
3s
[CV 4/5] END .....hidden_layer_sizes=(3, 4);, score=0.308 total time=   8.
6s
[CV 5/5] END .....hidden_layer_sizes=(3, 4);, score=0.236 total time=   7.
6s
[CV 1/5] END .....hidden_layer_sizes=(3, 5);, score=0.418 total time=   6.
7s
[CV 2/5] END .....hidden_layer_sizes=(3, 5);, score=0.544 total time=   6.
6s
[CV 3/5] END .....hidden_layer_sizes=(3, 5);, score=0.588 total time=   4.
6s
[CV 4/5] END .....hidden_layer_sizes=(3, 5);, score=0.220 total time=   8.
1s
[CV 5/5] END .....hidden_layer_sizes=(3, 5);, score=0.277 total time=   6.
7s
[CV 1/5] END .....hidden_layer_sizes=(3, 6);, score=0.415 total time=   5.
2s
[CV 2/5] END .....hidden_layer_sizes=(3, 6);, score=0.546 total time=   6.
0s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 3/5] END .....hidden_layer_sizes=(3, 6);, score=-6.489 total time= 1.2m
in
[CV 4/5] END .....hidden_layer_sizes=(3, 6);, score=0.343 total time=   7.
9s
[CV 5/5] END .....hidden_layer_sizes=(3, 6);, score=0.236 total time=   3.
7s
[CV 1/5] END .....hidden_layer_sizes=(3, 7);, score=0.553 total time=  10.
7s
[CV 2/5] END .....hidden_layer_sizes=(3, 7);, score=0.591 total time=   7.
5s
[CV 3/5] END .....hidden_layer_sizes=(3, 7);, score=0.631 total time=   7.
6s
[CV 4/5] END .....hidden_layer_sizes=(3, 7);, score=0.261 total time=   5.
6s
[CV 5/5] END .....hidden_layer_sizes=(3, 7);, score=0.357 total time=  10.
1s
[CV 1/5] END .....hidden_layer_sizes=(3, 8);, score=0.483 total time=   6.
0s
[CV 2/5] END .....hidden_layer_sizes=(3, 8);, score=0.603 total time=   8.
2s
[CV 3/5] END .....hidden_layer_sizes=(3, 8);, score=0.770 total time=   7.
9s
[CV 4/5] END .....hidden_layer_sizes=(3, 8);, score=0.222 total time=   6.
9s
[CV 5/5] END .....hidden_layer_sizes=(3, 8);, score=0.359 total time=  12.
5s
[CV 1/5] END .....hidden_layer_sizes=(3, 9);, score=0.415 total time=   8.
9s
[CV 2/5] END .....hidden_layer_sizes=(3, 9);, score=0.695 total time=  13.
3s
[CV 3/5] END .....hidden_layer_sizes=(3, 9);, score=0.630 total time=   7.
```

```
2s
[CV 4/5] END .........hidden_layer_sizes=(3, 9);, score=0.265 total time=   9.
0s
[CV 5/5] END .........hidden_layer_sizes=(3, 9);, score=0.361 total time=  13.
2s
[CV 1/5] END ........hidden_layer_sizes=(3, 10);, score=0.433 total time=  14.
0s
[CV 2/5] END ........hidden_layer_sizes=(3, 10);, score=0.548 total time=   5.
9s
[CV 3/5] END ........hidden_layer_sizes=(3, 10);, score=0.704 total time=   9.
7s
[CV 4/5] END ........hidden_layer_sizes=(3, 10);, score=0.312 total time=   7.
2s
[CV 5/5] END ........hidden_layer_sizes=(3, 10);, score=0.294 total time=  19.
0s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(4, 1);, score=-5.131 total time=  54.
6s
[CV 2/5] END .........hidden_layer_sizes=(4, 1);, score=0.601 total time=   9.
0s
[CV 3/5] END .........hidden_layer_sizes=(4, 1);, score=0.632 total time=  30.
9s
[CV 4/5] END .........hidden_layer_sizes=(4, 1);, score=0.221 total time=  11.
3s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 5/5] END ........hidden_layer_sizes=(4, 1);, score=-2.818 total time=  54.
6s
[CV 1/5] END .........hidden_layer_sizes=(4, 2);, score=0.414 total time=   6.
3s
[CV 2/5] END .........hidden_layer_sizes=(4, 2);, score=0.543 total time=   6.
8s
[CV 3/5] END .........hidden_layer_sizes=(4, 2);, score=0.591 total time=   7.
7s
[CV 4/5] END .........hidden_layer_sizes=(4, 2);, score=0.222 total time=   8.
2s
[CV 5/5] END .........hidden_layer_sizes=(4, 2);, score=0.335 total time=   8.
0s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 1/5] END ........hidden_layer_sizes=(4, 3);, score=-5.169 total time= 1.0m
in
[CV 2/5] END .........hidden_layer_sizes=(4, 3);, score=0.606 total time=   7.
7s
[CV 3/5] END .........hidden_layer_sizes=(4, 3);, score=0.642 total time=   5.
3s
[CV 4/5] END .........hidden_layer_sizes=(4, 3);, score=0.239 total time=   5.
8s
[CV 5/5] END .........hidden_layer_sizes=(4, 3);, score=0.274 total time=   4.
9s
[CV 1/5] END .........hidden_layer_sizes=(4, 4);, score=0.414 total time=   5.
1s
[CV 2/5] END .........hidden_layer_sizes=(4, 4);, score=0.538 total time=   4.
```

```
8s
[CV 3/5] END .........hidden_layer_sizes=(4, 4);, score=0.634 total time=   8.
2s
[CV 4/5] END .........hidden_layer_sizes=(4, 4);, score=0.340 total time=   6.
5s
[CV 5/5] END .........hidden_layer_sizes=(4, 4);, score=0.237 total time=   4.
9s
[CV 1/5] END .........hidden_layer_sizes=(4, 5);, score=0.507 total time=  14.
3s
[CV 2/5] END .........hidden_layer_sizes=(4, 5);, score=0.768 total time=  12.
7s
[CV 3/5] END .........hidden_layer_sizes=(4, 5);, score=0.804 total time=  22.
5s
[CV 4/5] END .........hidden_layer_sizes=(4, 5);, score=0.232 total time=   4.
8s
[CV 5/5] END .........hidden_layer_sizes=(4, 5);, score=0.277 total time=   5.
6s
[CV 1/5] END .........hidden_layer_sizes=(4, 6);, score=0.406 total time=   4.
4s
[CV 2/5] END .........hidden_layer_sizes=(4, 6);, score=0.604 total time=   5.
9s
[CV 3/5] END .........hidden_layer_sizes=(4, 6);, score=0.810 total time=   7.
4s
[CV 4/5] END .........hidden_layer_sizes=(4, 6);, score=0.233 total time=   7.
6s
[CV 5/5] END .........hidden_layer_sizes=(4, 6);, score=0.278 total time=   5.
5s
[CV 1/5] END .........hidden_layer_sizes=(4, 7);, score=0.419 total time=   7.
1s
[CV 2/5] END .........hidden_layer_sizes=(4, 7);, score=0.625 total time=   9.
5s
[CV 3/5] END .........hidden_layer_sizes=(4, 7);, score=0.589 total time=   4.
5s
[CV 4/5] END .........hidden_layer_sizes=(4, 7);, score=0.223 total time=   5.
3s
[CV 5/5] END .........hidden_layer_sizes=(4, 7);, score=0.238 total time=   4.
7s
[CV 1/5] END .........hidden_layer_sizes=(4, 8);, score=0.442 total time=   8.
5s
[CV 2/5] END .........hidden_layer_sizes=(4, 8);, score=0.591 total time=   7.
9s
[CV 3/5] END .........hidden_layer_sizes=(4, 8);, score=0.641 total time=   5.
2s
[CV 4/5] END .........hidden_layer_sizes=(4, 8);, score=0.358 total time=   9.
3s
[CV 5/5] END .........hidden_layer_sizes=(4, 8);, score=0.238 total time=   5.
0s
[CV 1/5] END .........hidden_layer_sizes=(4, 9);, score=0.446 total time=   3.
9s
[CV 2/5] END .........hidden_layer_sizes=(4, 9);, score=0.784 total time=   5.
8s
[CV 3/5] END .........hidden_layer_sizes=(4, 9);, score=0.587 total time=   8.
1s
[CV 4/5] END .........hidden_layer_sizes=(4, 9);, score=0.243 total time=   5.
4s
[CV 5/5] END .........hidden_layer_sizes=(4, 9);, score=0.238 total time=   4.
1s
[CV 1/5] END ........hidden_layer_sizes=(4, 10);, score=0.581 total time=   6.
4s
[CV 2/5] END ........hidden_layer_sizes=(4, 10);, score=0.545 total time=   6.
1s
```

```
1s
[CV 3/5] END ........hidden_layer_sizes=(4, 10);, score=0.801 total time=   6.
9s
[CV 4/5] END ........hidden_layer_sizes=(4, 10);, score=0.235 total time=  15.
9s
[CV 5/5] END ........hidden_layer_sizes=(4, 10);, score=0.277 total time=   5.
7s
[CV 1/5] END .........hidden_layer_sizes=(5, 1);, score=0.480 total time=   6.
9s
[CV 2/5] END .........hidden_layer_sizes=(5, 1);, score=0.544 total time=   5.
5s
[CV 3/5] END .........hidden_layer_sizes=(5, 1);, score=0.588 total time=  18.
9s
[CV 4/5] END .........hidden_layer_sizes=(5, 1);, score=0.220 total time=   6.
1s
[CV 5/5] END .........hidden_layer_sizes=(5, 1);, score=0.275 total time=   9.
3s
[CV 1/5] END .........hidden_layer_sizes=(5, 2);, score=0.472 total time=   6.
7s
[CV 2/5] END .........hidden_layer_sizes=(5, 2);, score=0.543 total time=  21.
1s
[CV 3/5] END .........hidden_layer_sizes=(5, 2);, score=0.581 total time=   6.
8s
[CV 4/5] END .........hidden_layer_sizes=(5, 2);, score=0.220 total time=   6.
2s
[CV 5/5] END .........hidden_layer_sizes=(5, 2);, score=0.235 total time=  22.
1s
[CV 1/5] END .........hidden_layer_sizes=(5, 3);, score=0.474 total time=   5.
3s
[CV 2/5] END .........hidden_layer_sizes=(5, 3);, score=0.786 total time=  10.
8s
[CV 3/5] END .........hidden_layer_sizes=(5, 3);, score=0.624 total time=   7.
7s
[CV 4/5] END .........hidden_layer_sizes=(5, 3);, score=0.345 total time=  17.
4s
[CV 5/5] END .........hidden_layer_sizes=(5, 3);, score=0.236 total time=   4.
4s
[CV 1/5] END .........hidden_layer_sizes=(5, 4);, score=0.412 total time=   4.
6s
[CV 2/5] END .........hidden_layer_sizes=(5, 4);, score=0.768 total time=   8.
1s
[CV 3/5] END .........hidden_layer_sizes=(5, 4);, score=0.819 total time=   5.
7s
[CV 4/5] END .........hidden_layer_sizes=(5, 4);, score=0.222 total time=   5.
6s
[CV 5/5] END .........hidden_layer_sizes=(5, 4);, score=0.355 total time=  14.
0s
[CV 1/5] END .........hidden_layer_sizes=(5, 5);, score=0.565 total time=   9.
8s
[CV 2/5] END .........hidden_layer_sizes=(5, 5);, score=0.602 total time=   6.
1s
[CV 3/5] END .........hidden_layer_sizes=(5, 5);, score=0.641 total time=   6.
0s
[CV 4/5] END .........hidden_layer_sizes=(5, 5);, score=0.212 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(5, 5);, score=0.352 total time=  13.
4s
[CV 1/5] END .........hidden_layer_sizes=(5, 6);, score=0.458 total time=   5.
4s
[CV 2/5] END .........hidden_layer_sizes=(5, 6);, score=0.603 total time=   6.
4s
```

```
[CV 3/5] END .........hidden_layer_sizes=(5, 6);, score=0.808 total time=  12.
1s
[CV 4/5] END .........hidden_layer_sizes=(5, 6);, score=0.361 total time=  12.
6s
[CV 5/5] END .........hidden_layer_sizes=(5, 6);, score=0.237 total time=   3.
9s
[CV 1/5] END .........hidden_layer_sizes=(5, 7);, score=0.562 total time=   8.
5s
[CV 2/5] END .........hidden_layer_sizes=(5, 7);, score=0.604 total time=   6.
5s
[CV 3/5] END .........hidden_layer_sizes=(5, 7);, score=0.698 total time=   7.
6s
[CV 4/5] END .........hidden_layer_sizes=(5, 7);, score=0.321 total time=  13.
4s
[CV 5/5] END .........hidden_layer_sizes=(5, 7);, score=0.272 total time=   5.
1s
[CV 1/5] END .........hidden_layer_sizes=(5, 8);, score=0.590 total time=   7.
6s
[CV 2/5] END .........hidden_layer_sizes=(5, 8);, score=0.595 total time=   6.
0s
[CV 3/5] END .........hidden_layer_sizes=(5, 8);, score=0.827 total time=  13.
8s
[CV 4/5] END .........hidden_layer_sizes=(5, 8);, score=0.331 total time=   5.
4s
[CV 5/5] END .........hidden_layer_sizes=(5, 8);, score=0.238 total time=   3.
7s
[CV 1/5] END .........hidden_layer_sizes=(5, 9);, score=0.416 total time=   4.
8s
[CV 2/5] END .........hidden_layer_sizes=(5, 9);, score=0.776 total time=   9.
6s
[CV 3/5] END .........hidden_layer_sizes=(5, 9);, score=0.621 total time=   5.
5s
[CV 4/5] END .........hidden_layer_sizes=(5, 9);, score=0.249 total time=   4.
1s
[CV 5/5] END .........hidden_layer_sizes=(5, 9);, score=0.367 total time=   8.
5s
[CV 1/5] END ........hidden_layer_sizes=(5, 10);, score=0.592 total time=   6.
6s
[CV 2/5] END ........hidden_layer_sizes=(5, 10);, score=0.592 total time=   5.
2s
[CV 3/5] END ........hidden_layer_sizes=(5, 10);, score=0.582 total time=   5.
7s
[CV 4/5] END ........hidden_layer_sizes=(5, 10);, score=0.269 total time=  13.
7s
[CV 5/5] END ........hidden_layer_sizes=(5, 10);, score=0.343 total time=  10.
4s
[CV 1/5] END .........hidden_layer_sizes=(6, 1);, score=0.481 total time=  23.
6s
[CV 2/5] END .........hidden_layer_sizes=(6, 1);, score=0.595 total time=   7.
8s
[CV 3/5] END .........hidden_layer_sizes=(6, 1);, score=0.586 total time=  27.
8s
[CV 4/5] END .........hidden_layer_sizes=(6, 1);, score=0.233 total time=  22.
7s
[CV 5/5] END .........hidden_layer_sizes=(6, 1);, score=0.237 total time=   5.
8s
[CV 1/5] END .........hidden_layer_sizes=(6, 2);, score=0.478 total time=   6.
3s
[CV 2/5] END .........hidden_layer_sizes=(6, 2);, score=0.778 total time=  18.
9s
```

```
[CV 3/5] END .........hidden_layer_sizes=(6, 2);, score=0.643 total time=   6.
4s
[CV 4/5] END .........hidden_layer_sizes=(6, 2);, score=0.242 total time=   6.
4s
[CV 5/5] END .........hidden_layer_sizes=(6, 2);, score=0.359 total time=  11.
0s
[CV 1/5] END .........hidden_layer_sizes=(6, 3);, score=0.419 total time=   5.
1s
[CV 2/5] END .........hidden_layer_sizes=(6, 3);, score=0.767 total time=  17.
8s
[CV 3/5] END .........hidden_layer_sizes=(6, 3);, score=0.797 total time=  28.
8s
[CV 4/5] END .........hidden_layer_sizes=(6, 3);, score=0.222 total time=   4.
9s
[CV 5/5] END .........hidden_layer_sizes=(6, 3);, score=0.279 total time=   5.
0s
[CV 1/5] END .........hidden_layer_sizes=(6, 4);, score=0.597 total time=  10.
0s
[CV 2/5] END .........hidden_layer_sizes=(6, 4);, score=0.607 total time=   5.
0s
[CV 3/5] END .........hidden_layer_sizes=(6, 4);, score=0.587 total time=   4.
4s
[CV 4/5] END .........hidden_layer_sizes=(6, 4);, score=0.359 total time=  10.
3s
[CV 5/5] END .........hidden_layer_sizes=(6, 4);, score=0.277 total time=   5.
1s
[CV 1/5] END .........hidden_layer_sizes=(6, 5);, score=0.405 total time=   5.
2s
[CV 2/5] END .........hidden_layer_sizes=(6, 5);, score=0.874 total time=  14.
5s
[CV 3/5] END .........hidden_layer_sizes=(6, 5);, score=0.817 total time=   6.
8s
[CV 4/5] END .........hidden_layer_sizes=(6, 5);, score=0.244 total time=   5.
3s
[CV 5/5] END .........hidden_layer_sizes=(6, 5);, score=0.233 total time=   4.
5s
[CV 1/5] END .........hidden_layer_sizes=(6, 6);, score=0.573 total time=   9.
2s
[CV 2/5] END .........hidden_layer_sizes=(6, 6);, score=0.597 total time=   4.
8s
[CV 3/5] END .........hidden_layer_sizes=(6, 6);, score=0.574 total time=   4.
1s
[CV 4/5] END .........hidden_layer_sizes=(6, 6);, score=0.240 total time=   6.
9s
[CV 5/5] END .........hidden_layer_sizes=(6, 6);, score=0.237 total time=   5.
1s
[CV 1/5] END .........hidden_layer_sizes=(6, 7);, score=0.590 total time=   9.
5s
[CV 2/5] END .........hidden_layer_sizes=(6, 7);, score=0.785 total time=   8.
5s
[CV 3/5] END .........hidden_layer_sizes=(6, 7);, score=0.571 total time=   4.
7s
[CV 4/5] END .........hidden_layer_sizes=(6, 7);, score=0.346 total time=  11.
5s
[CV 5/5] END .........hidden_layer_sizes=(6, 7);, score=0.276 total time=   4.
3s
[CV 1/5] END .........hidden_layer_sizes=(6, 8);, score=0.412 total time=   4.
6s
[CV 2/5] END .........hidden_layer_sizes=(6, 8);, score=0.869 total time=  12.
7s
[CV 3/5] END .........hidden_layer_sizes=(6, 8);, score=0.788 total time=   8.
```

```
[CV 3/5] END .........hidden_layer_sizes=(6, 8);, score=0.788 total time=   8.
4s
[CV 4/5] END .........hidden_layer_sizes=(6, 8);, score=0.220 total time=   6.
3s
[CV 5/5] END .........hidden_layer_sizes=(6, 8);, score=0.238 total time=   4.
3s
[CV 1/5] END .........hidden_layer_sizes=(6, 9);, score=0.589 total time=   8.
6s
[CV 2/5] END .........hidden_layer_sizes=(6, 9);, score=0.534 total time=   5.
2s
[CV 3/5] END .........hidden_layer_sizes=(6, 9);, score=0.817 total time=  11.
7s
[CV 4/5] END .........hidden_layer_sizes=(6, 9);, score=0.222 total time=   4.
9s
[CV 5/5] END .........hidden_layer_sizes=(6, 9);, score=0.238 total time=   3.
9s
[CV 1/5] END ........hidden_layer_sizes=(6, 10);, score=0.587 total time=   8.
1s
[CV 2/5] END ........hidden_layer_sizes=(6, 10);, score=0.539 total time=   3.
3s
[CV 3/5] END ........hidden_layer_sizes=(6, 10);, score=0.580 total time=   6.
7s
[CV 4/5] END ........hidden_layer_sizes=(6, 10);, score=0.225 total time=   4.
0s
[CV 5/5] END ........hidden_layer_sizes=(6, 10);, score=0.275 total time=   4.
8s
[CV 1/5] END .........hidden_layer_sizes=(7, 1);, score=0.470 total time=  26.
1s
[CV 2/5] END .........hidden_layer_sizes=(7, 1);, score=0.596 total time=   6.
5s
[CV 3/5] END .........hidden_layer_sizes=(7, 1);, score=0.588 total time=  24.
9s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 4/5] END .........hidden_layer_sizes=(7, 1);, score=-2.734 total time=  59.
9s
[CV 5/5] END .........hidden_layer_sizes=(7, 1);, score=0.276 total time=   5.
9s
[CV 1/5] END .........hidden_layer_sizes=(7, 2);, score=0.476 total time=   6.
5s
[CV 2/5] END .........hidden_layer_sizes=(7, 2);, score=0.598 total time=   5.
4s
[CV 3/5] END .........hidden_layer_sizes=(7, 2);, score=0.804 total time=  10.
0s
[CV 4/5] END .........hidden_layer_sizes=(7, 2);, score=0.242 total time=   5.
5s
[CV 5/5] END .........hidden_layer_sizes=(7, 2);, score=0.236 total time=   5.
1s
[CV 1/5] END .........hidden_layer_sizes=(7, 3);, score=0.479 total time=  16.
2s
[CV 2/5] END .........hidden_layer_sizes=(7, 3);, score=0.603 total time=   6.
1s
[CV 3/5] END .........hidden_layer_sizes=(7, 3);, score=0.640 total time=   8.
4s
[CV 4/5] END .........hidden_layer_sizes=(7, 3);, score=0.349 total time=  12.
6s
[CV 5/5] END .........hidden_layer_sizes=(7, 3);, score=0.275 total time=   5.
7s
[CV 1/5] END .........hidden_layer_sizes=(7, 4);, score=0.471 total time=   6.
```

```
2s
[CV 2/5] END .........hidden_layer_sizes=(7, 4);, score=0.544 total time=   6.
2s
[CV 3/5] END .........hidden_layer_sizes=(7, 4);, score=0.639 total time=   5.
4s
[CV 4/5] END .........hidden_layer_sizes=(7, 4);, score=0.240 total time=   6.
1s
[CV 5/5] END .........hidden_layer_sizes=(7, 4);, score=0.279 total time=   5.
0s
[CV 1/5] END .........hidden_layer_sizes=(7, 5);, score=0.479 total time=   4.
2s
[CV 2/5] END .........hidden_layer_sizes=(7, 5);, score=0.599 total time=   4.
9s
[CV 3/5] END .........hidden_layer_sizes=(7, 5);, score=0.637 total time=   4.
9s
[CV 4/5] END .........hidden_layer_sizes=(7, 5);, score=0.246 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(7, 5);, score=0.359 total time=  10.
6s
[CV 1/5] END .........hidden_layer_sizes=(7, 6);, score=0.485 total time=   6.
5s
[CV 2/5] END .........hidden_layer_sizes=(7, 6);, score=0.605 total time=   6.
4s
[CV 3/5] END .........hidden_layer_sizes=(7, 6);, score=0.576 total time=   5.
1s
[CV 4/5] END .........hidden_layer_sizes=(7, 6);, score=0.252 total time=   7.
7s
[CV 5/5] END .........hidden_layer_sizes=(7, 6);, score=0.279 total time=   5.
3s
[CV 1/5] END .........hidden_layer_sizes=(7, 7);, score=0.408 total time=   5.
0s
[CV 2/5] END .........hidden_layer_sizes=(7, 7);, score=0.607 total time=   5.
4s
[CV 3/5] END .........hidden_layer_sizes=(7, 7);, score=0.644 total time=   4.
8s
[CV 4/5] END .........hidden_layer_sizes=(7, 7);, score=0.262 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(7, 7);, score=0.236 total time=   4.
6s
[CV 1/5] END .........hidden_layer_sizes=(7, 8);, score=0.485 total time=   4.
8s
[CV 2/5] END .........hidden_layer_sizes=(7, 8);, score=0.542 total time=   3.
6s
[CV 3/5] END .........hidden_layer_sizes=(7, 8);, score=0.579 total time=   4.
5s
[CV 4/5] END .........hidden_layer_sizes=(7, 8);, score=0.224 total time=   4.
5s
[CV 5/5] END .........hidden_layer_sizes=(7, 8);, score=0.274 total time=   4.
1s
[CV 1/5] END .........hidden_layer_sizes=(7, 9);, score=0.480 total time=   4.
5s
[CV 2/5] END .........hidden_layer_sizes=(7, 9);, score=0.775 total time=   8.
0s
[CV 3/5] END .........hidden_layer_sizes=(7, 9);, score=0.758 total time=   8.
4s
[CV 4/5] END .........hidden_layer_sizes=(7, 9);, score=0.331 total time=   7.
0s
[CV 5/5] END .........hidden_layer_sizes=(7, 9);, score=0.357 total time=   8.
5s
[CV 1/5] END ........hidden_layer_sizes=(7, 10);, score=0.474 total time=   4.
```

```
7s
[CV 2/5] END ........hidden_layer_sizes=(7, 10);, score=0.761 total time=   5.
6s
[CV 3/5] END ........hidden_layer_sizes=(7, 10);, score=0.769 total time=  10.
3s
[CV 4/5] END ........hidden_layer_sizes=(7, 10);, score=0.344 total time=   9.
4s
[CV 5/5] END ........hidden_layer_sizes=(7, 10);, score=0.302 total time=   5.
7s
[CV 1/5] END .........hidden_layer_sizes=(8, 1);, score=0.414 total time=  12.
0s
[CV 2/5] END .........hidden_layer_sizes=(8, 1);, score=0.602 total time=   6.
0s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 3/5] END .........hidden_layer_sizes=(8, 1);, score=-6.472 total time=  58.
1s
[CV 4/5] END .........hidden_layer_sizes=(8, 1);, score=0.225 total time=  13.
3s
[CV 5/5] END .........hidden_layer_sizes=(8, 1);, score=0.237 total time=   6.
0s
[CV 1/5] END .........hidden_layer_sizes=(8, 2);, score=0.503 total time=   9.
8s
[CV 2/5] END .........hidden_layer_sizes=(8, 2);, score=0.586 total time=  26.
6s
[CV 3/5] END .........hidden_layer_sizes=(8, 2);, score=0.629 total time=   9.
1s
[CV 4/5] END .........hidden_layer_sizes=(8, 2);, score=0.331 total time=  11.
9s
[CV 5/5] END .........hidden_layer_sizes=(8, 2);, score=0.276 total time=   5.
3s
[CV 1/5] END .........hidden_layer_sizes=(8, 3);, score=0.482 total time=   4.
9s
[CV 2/5] END .........hidden_layer_sizes=(8, 3);, score=0.781 total time=   8.
2s
[CV 3/5] END .........hidden_layer_sizes=(8, 3);, score=0.802 total time=   9.
3s
[CV 4/5] END .........hidden_layer_sizes=(8, 3);, score=0.237 total time=   5.
1s
[CV 5/5] END .........hidden_layer_sizes=(8, 3);, score=0.352 total time=  13.
1s
[CV 1/5] END .........hidden_layer_sizes=(8, 4);, score=0.591 total time=   8.
8s
[CV 2/5] END .........hidden_layer_sizes=(8, 4);, score=0.734 total time=   9.
5s
[CV 3/5] END .........hidden_layer_sizes=(8, 4);, score=0.812 total time=   9.
5s
[CV 4/5] END .........hidden_layer_sizes=(8, 4);, score=0.239 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(8, 4);, score=0.360 total time=  10.
8s
[CV 1/5] END .........hidden_layer_sizes=(8, 5);, score=0.479 total time=   5.
2s
[CV 2/5] END .........hidden_layer_sizes=(8, 5);, score=0.783 total time=   9.
0s
[CV 3/5] END .........hidden_layer_sizes=(8, 5);, score=0.785 total time=   6.
1s
[CV 4/5] END .........hidden_layer_sizes=(8, 5);, score=0.243 total time=   5.
```

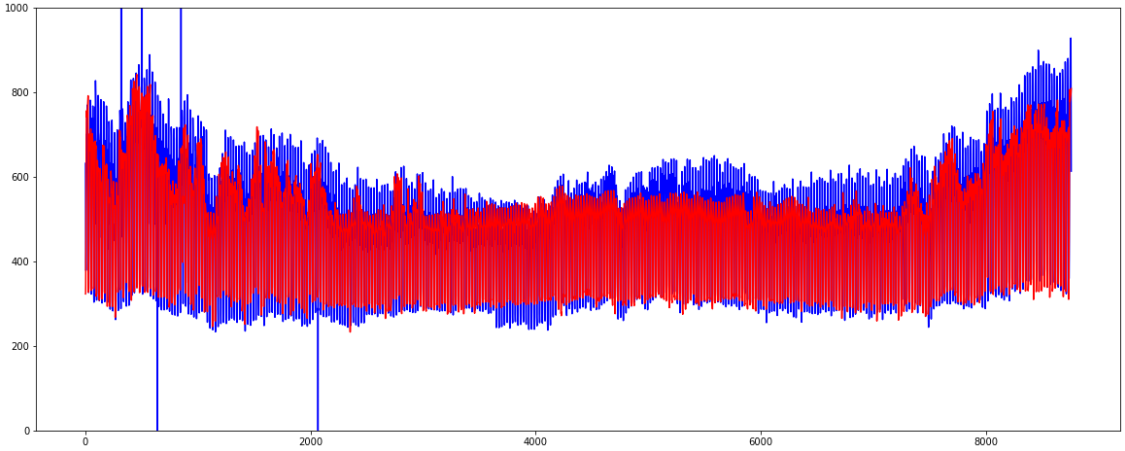
```
1s
[CV 5/5] END .........hidden_layer_sizes=(8, 5);, score=0.353 total time=  10.
4s
[CV 1/5] END .........hidden_layer_sizes=(8, 6);, score=0.414 total time=   5.
4s
[CV 2/5] END .........hidden_layer_sizes=(8, 6);, score=0.769 total time=  11.
1s
[CV 3/5] END .........hidden_layer_sizes=(8, 6);, score=0.814 total time=  12.
2s
[CV 4/5] END .........hidden_layer_sizes=(8, 6);, score=0.239 total time=   5.
5s
[CV 5/5] END .........hidden_layer_sizes=(8, 6);, score=0.237 total time=   4.
3s
[CV 1/5] END .........hidden_layer_sizes=(8, 7);, score=0.591 total time=   6.
7s
[CV 2/5] END .........hidden_layer_sizes=(8, 7);, score=0.543 total time=   6.
4s
[CV 3/5] END .........hidden_layer_sizes=(8, 7);, score=0.589 total time=   3.
6s
[CV 4/5] END .........hidden_layer_sizes=(8, 7);, score=0.344 total time=   5.
5s
[CV 5/5] END .........hidden_layer_sizes=(8, 7);, score=0.269 total time=   6.
8s
[CV 1/5] END .........hidden_layer_sizes=(8, 8);, score=0.412 total time=   4.
0s
[CV 2/5] END .........hidden_layer_sizes=(8, 8);, score=0.803 total time=   8.
2s
[CV 3/5] END .........hidden_layer_sizes=(8, 8);, score=0.584 total time=   4.
1s
[CV 4/5] END .........hidden_layer_sizes=(8, 8);, score=0.224 total time=   4.
3s
[CV 5/5] END .........hidden_layer_sizes=(8, 8);, score=0.360 total time=   7.
6s
[CV 1/5] END .........hidden_layer_sizes=(8, 9);, score=0.442 total time=   4.
2s
[CV 2/5] END .........hidden_layer_sizes=(8, 9);, score=0.600 total time=   5.
0s
[CV 3/5] END .........hidden_layer_sizes=(8, 9);, score=0.637 total time=   4.
6s
[CV 4/5] END .........hidden_layer_sizes=(8, 9);, score=0.332 total time=   8.
5s
[CV 5/5] END .........hidden_layer_sizes=(8, 9);, score=0.236 total time=   4.
6s
[CV 1/5] END ........hidden_layer_sizes=(8, 10);, score=0.403 total time=   3.
9s
[CV 2/5] END ........hidden_layer_sizes=(8, 10);, score=0.538 total time=   4.
2s
[CV 3/5] END ........hidden_layer_sizes=(8, 10);, score=0.795 total time=   7.
3s
[CV 4/5] END ........hidden_layer_sizes=(8, 10);, score=0.318 total time=   7.
8s
[CV 5/5] END ........hidden_layer_sizes=(8, 10);, score=0.357 total time=  10.
6s
[CV 1/5] END .........hidden_layer_sizes=(9, 1);, score=0.415 total time=  35.
6s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 2/5] END .........hidden_layer_sizes=(9, 1);, score=-6.754 total time= 1.1m
in
```

```
[CV 3/5] END .........hidden_layer_sizes=(9, 1);, score=0.587 total time=  13.
4s
[CV 4/5] END .........hidden_layer_sizes=(9, 1);, score=0.222 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(9, 1);, score=0.276 total time=   6.
0s
[CV 1/5] END .........hidden_layer_sizes=(9, 2);, score=0.588 total time=  11.
9s
[CV 2/5] END .........hidden_layer_sizes=(9, 2);, score=0.544 total time=  33.
1s
[CV 3/5] END .........hidden_layer_sizes=(9, 2);, score=0.627 total time=   6.
3s
[CV 4/5] END .........hidden_layer_sizes=(9, 2);, score=0.225 total time=   5.
7s
[CV 5/5] END .........hidden_layer_sizes=(9, 2);, score=0.277 total time=   6.
5s
[CV 1/5] END .........hidden_layer_sizes=(9, 3);, score=0.416 total time=  31.
8s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 2/5] END .........hidden_layer_sizes=(9, 3);, score=-6.757 total time= 1.3m
in
[CV 3/5] END .........hidden_layer_sizes=(9, 3);, score=0.646 total time=  19.
1s
[CV 4/5] END .........hidden_layer_sizes=(9, 3);, score=0.223 total time=  37.
2s
[CV 5/5] END .........hidden_layer_sizes=(9, 3);, score=0.266 total time=   4.
7s
[CV 1/5] END .........hidden_layer_sizes=(9, 4);, score=0.411 total time=   5.
2s
[CV 2/5] END .........hidden_layer_sizes=(9, 4);, score=0.601 total time=  25.
0s
[CV 3/5] END .........hidden_layer_sizes=(9, 4);, score=0.792 total time=   9.
0s
[CV 4/5] END .........hidden_layer_sizes=(9, 4);, score=0.333 total time=  14.
3s
[CV 5/5] END .........hidden_layer_sizes=(9, 4);, score=0.277 total time=   4.
4s
[CV 1/5] END .........hidden_layer_sizes=(9, 5);, score=0.621 total time=  11.
8s
[CV 2/5] END .........hidden_layer_sizes=(9, 5);, score=0.545 total time=   5.
9s
[CV 3/5] END .........hidden_layer_sizes=(9, 5);, score=0.639 total time=   4.
9s
[CV 4/5] END .........hidden_layer_sizes=(9, 5);, score=0.307 total time=   9.
5s
[CV 5/5] END .........hidden_layer_sizes=(9, 5);, score=0.366 total time=   8.
5s
[CV 1/5] END .........hidden_layer_sizes=(9, 6);, score=0.412 total time=   5.
4s
[CV 2/5] END .........hidden_layer_sizes=(9, 6);, score=0.597 total time=   4.
4s
[CV 3/5] END .........hidden_layer_sizes=(9, 6);, score=0.812 total time=   9.
4s
[CV 4/5] END .........hidden_layer_sizes=(9, 6);, score=0.224 total time=   4.
7s
[CV 5/5] END .........hidden_layer_sizes=(9, 6);, score=0.277 total time=   3.
9s
```

```
[CV 1/5] END .........hidden_layer_sizes=(9, 7);, score=0.559 total time=   7.
2s
[CV 2/5] END .........hidden_layer_sizes=(9, 7);, score=0.602 total time=   5.
6s
[CV 3/5] END .........hidden_layer_sizes=(9, 7);, score=0.639 total time=   4.
4s
[CV 4/5] END .........hidden_layer_sizes=(9, 7);, score=0.222 total time=   4.
0s
[CV 5/5] END .........hidden_layer_sizes=(9, 7);, score=0.363 total time=   7.
6s
[CV 1/5] END .........hidden_layer_sizes=(9, 8);, score=0.593 total time=   5.
9s
[CV 2/5] END .........hidden_layer_sizes=(9, 8);, score=0.774 total time=  10.
6s
[CV 3/5] END .........hidden_layer_sizes=(9, 8);, score=0.854 total time=  11.
4s
[CV 4/5] END .........hidden_layer_sizes=(9, 8);, score=0.337 total time=   9.
7s
[CV 5/5] END .........hidden_layer_sizes=(9, 8);, score=0.237 total time=   3.
6s
[CV 1/5] END .........hidden_layer_sizes=(9, 9);, score=0.582 total time=   6.
9s
[CV 2/5] END .........hidden_layer_sizes=(9, 9);, score=0.783 total time=   7.
3s
[CV 3/5] END .........hidden_layer_sizes=(9, 9);, score=0.814 total time=   6.
2s
[CV 4/5] END .........hidden_layer_sizes=(9, 9);, score=0.342 total time=   8.
5s
[CV 5/5] END .........hidden_layer_sizes=(9, 9);, score=0.363 total time=   9.
3s
[CV 1/5] END ........hidden_layer_sizes=(9, 10);, score=0.470 total time=   4.
5s
[CV 2/5] END ........hidden_layer_sizes=(9, 10);, score=0.542 total time=   4.
0s
[CV 3/5] END ........hidden_layer_sizes=(9, 10);, score=0.582 total time=   6.
8s
[CV 4/5] END ........hidden_layer_sizes=(9, 10);, score=0.324 total time=   5.
9s
[CV 5/5] END ........hidden_layer_sizes=(9, 10);, score=0.336 total time=   9.
7s
[CV 1/5] END ........hidden_layer_sizes=(10, 1);, score=0.463 total time=  22.
6s
[CV 2/5] END ........hidden_layer_sizes=(10, 1);, score=0.599 total time=  32.
6s
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 3/5] END .......hidden_layer_sizes=(10, 1);, score=-6.533 total time= 1.2m
in
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
[CV 4/5] END .......hidden_layer_sizes=(10, 1);, score=-2.708 total time= 1.1m
in
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perc
eptron.py:696: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (5
000) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
[CV 5/5] END .......hidden_layer_sizes=(10, 1);, score=-2.824 total time= 1.0m
in
[CV 1/5] END ........hidden_layer_sizes=(10, 2);, score=0.467 total time=   5.
0s
[CV 2/5] END ........hidden_layer_sizes=(10, 2);, score=0.543 total time=   5.
2s
[CV 3/5] END ........hidden_layer_sizes=(10, 2);, score=0.631 total time=   6.
0s
[CV 4/5] END ........hidden_layer_sizes=(10, 2);, score=0.248 total time=   6.
5s
[CV 5/5] END ........hidden_layer_sizes=(10, 2);, score=0.355 total time=  12.
6s
[CV 1/5] END ........hidden_layer_sizes=(10, 3);, score=0.600 total time=   9.
9s
[CV 2/5] END ........hidden_layer_sizes=(10, 3);, score=0.600 total time=   4.
9s
[CV 3/5] END ........hidden_layer_sizes=(10, 3);, score=0.812 total time=  11.
1s
[CV 4/5] END ........hidden_layer_sizes=(10, 3);, score=0.246 total time=   8.
3s
[CV 5/5] END ........hidden_layer_sizes=(10, 3);, score=0.238 total time=  36.
0s
[CV 1/5] END ........hidden_layer_sizes=(10, 4);, score=0.585 total time=  11.
2s
[CV 2/5] END ........hidden_layer_sizes=(10, 4);, score=0.749 total time=   6.
2s
[CV 3/5] END ........hidden_layer_sizes=(10, 4);, score=0.635 total time=   6.
4s
[CV 4/5] END ........hidden_layer_sizes=(10, 4);, score=0.238 total time=   4.
5s
[CV 5/5] END ........hidden_layer_sizes=(10, 4);, score=0.360 total time=  10.
9s
[CV 1/5] END ........hidden_layer_sizes=(10, 5);, score=0.584 total time=   6.
9s
[CV 2/5] END ........hidden_layer_sizes=(10, 5);, score=0.754 total time=   7.
0s
[CV 3/5] END ........hidden_layer_sizes=(10, 5);, score=0.585 total time=   4.
1s
[CV 4/5] END ........hidden_layer_sizes=(10, 5);, score=0.298 total time=   9.
4s
[CV 5/5] END ........hidden_layer_sizes=(10, 5);, score=0.278 total time=   4.
9s
[CV 1/5] END ........hidden_layer_sizes=(10, 6);, score=0.573 total time=   8.
6s
[CV 2/5] END ........hidden_layer_sizes=(10, 6);, score=0.794 total time=   6.
7s
[CV 3/5] END ........hidden_layer_sizes=(10, 6);, score=0.799 total time=   8.
3s
[CV 4/5] END ........hidden_layer_sizes=(10, 6);, score=0.330 total time=   5.
9s
[CV 5/5] END ........hidden_layer_sizes=(10, 6);, score=0.279 total time=   4.
2s
[CV 1/5] END ........hidden_layer_sizes=(10, 7);, score=0.558 total time=   9.
2s
[CV 2/5] END ........hidden_layer_sizes=(10, 7);, score=0.783 total time=   6.
1s
[CV 3/5] END ........hidden_layer_sizes=(10, 7);, score=0.821 total time=   9.
0s
[CV 4/5] END ........hidden_layer_sizes=(10, 7);, score=0.319 total time=   5.
3s
[CV 5/5] END ........hidden_layer_sizes=(10, 7);, score=0.272 total time=   3
```

```
[CV 5/5] END ........hidden_layer_sizes=(10, 7);, score=0.272 total time=   3.
6s
[CV 1/5] END ........hidden_layer_sizes=(10, 8);, score=0.566 total time=   7.
1s
[CV 2/5] END ........hidden_layer_sizes=(10, 8);, score=0.538 total time=   5.
2s
[CV 3/5] END ........hidden_layer_sizes=(10, 8);, score=0.851 total time=   9.
7s
[CV 4/5] END ........hidden_layer_sizes=(10, 8);, score=0.292 total time=   4.
7s
[CV 5/5] END ........hidden_layer_sizes=(10, 8);, score=0.279 total time=   3.
6s
[CV 1/5] END ........hidden_layer_sizes=(10, 9);, score=0.590 total time=  12.
5s
[CV 2/5] END ........hidden_layer_sizes=(10, 9);, score=0.779 total time=   7.
1s
[CV 3/5] END ........hidden_layer_sizes=(10, 9);, score=0.631 total time=   4.
5s
[CV 4/5] END ........hidden_layer_sizes=(10, 9);, score=0.316 total time=   7.
3s
[CV 5/5] END ........hidden_layer_sizes=(10, 9);, score=0.350 total time=   5.
4s
[CV 1/5] END .......hidden_layer_sizes=(10, 10);, score=0.416 total time=   3.
3s
[CV 2/5] END .......hidden_layer_sizes=(10, 10);, score=0.575 total time=   5.
3s
[CV 3/5] END .......hidden_layer_sizes=(10, 10);, score=0.571 total time=   4.
5s
[CV 4/5] END .......hidden_layer_sizes=(10, 10);, score=0.247 total time=   4.
5s
[CV 5/5] END .......hidden_layer_sizes=(10, 10);, score=0.365 total time=   7.
5s
Grid best parameter (max. R2):  {'hidden_layer_sizes': (9, 9)}
Grid best score (R2):  0.5767142177992696
R2 Score on test: 0.7896391321596805
```




```
Final R2 Score: 0.5542847245302072
```

Based on the models above, we obtained that the best method to predict the energy was a Random Forest with 70 Trees using the following variables: 'temp_per_day_2005','HourOfDay','full_humid_2005'

- Mean energy per Day
- Hour of the Day
- Humidity of the Hour

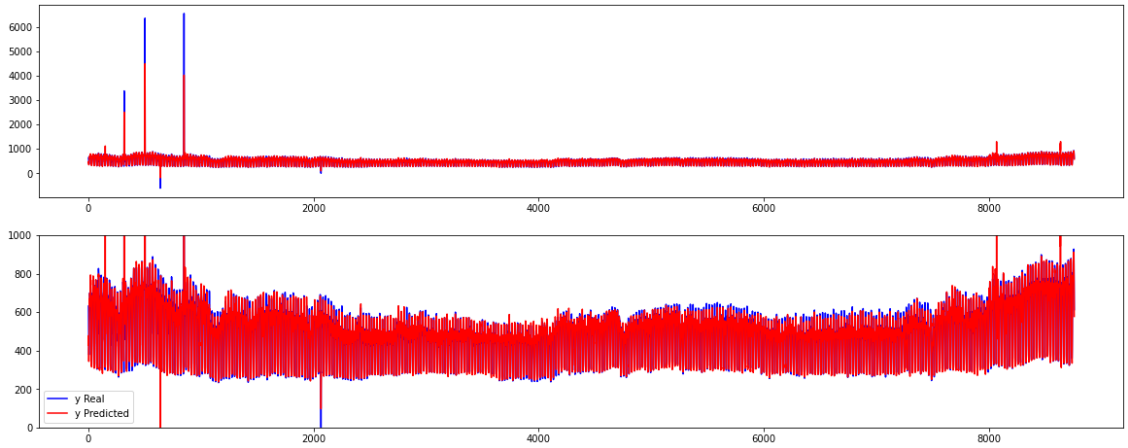
Hence, we obtained 2 different models to predict energy, each one for a different year

02.5 Final Model for 2005

In [ ]:
```
X_2005= df_energy[['temp_per_day_2005','HourOfDay','full_humid_2005']]
y_2005= df_energy['energy_consumpt_2005']

X_train, X_test, y_train, y_test= train_test_split(X_2005,y_2005,random_state
```

In [ ]:
```
parameters= {"n_estimators":list(range(10,101,10))}
rf = RandomForestRegressor(random_state=0, n_estimators=70)
rf.fit(X_train,y_train)

print("R2 Score on test:",r2_score(y_test, rf.predict(X_test)))
fig, axs= plt.subplots(2,1,figsize=(20,8))
axs[0]= plotRgresion(y_2005,rf.predict(X_2005),axs[0],False,0,1000)
axs[1]= plotRgresion(y_2005,rf.predict(X_2005),axs[1],True,0,1000)
plt.legend()
plt.show()
print("Final R2 Score:",r2_score(y_2005,rf.predict(X_2005)))
```

```
R2 Score on test: 0.6105817867373805
```




```
Final R2 Score: 0.8696535631662909
```

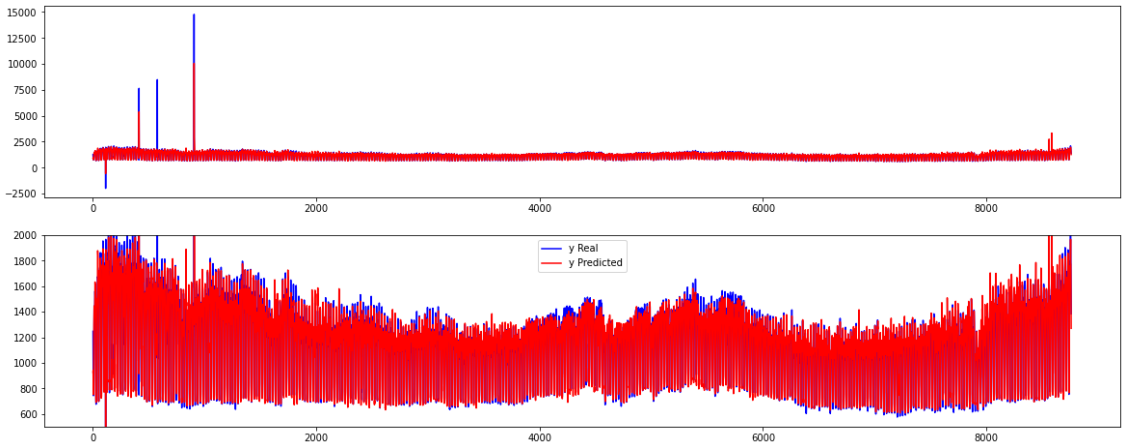
02.6 Final Model for 2006

In [ ]:
```
X_2006= df_energy[['temp_per_day_2006','HourOfDay','full_humid_2006']]
y_2006= df_energy['energy_consumpt_2006']

X_train, X_test, y_train, y_test= train_test_split(X_2006,y_2006,random_state
```

In [ ]:
```
parameters= {"n_estimators":list(range(10,101,10))}
rf = RandomForestRegressor(random_state=0, n_estimators=70)
rf.fit(X_train,y_train)

print("R2 Score on test:",r2_score(y_test, rf.predict(X_test)))
fig, axs= plt.subplots(2,1,figsize=(20,8))
axs[0]= plotRgresion(y_2006,rf.predict(X_2006),axs[0],False,0,1000)
axs[1]= plotRgresion(y_2006,rf.predict(X_2006),axs[1],True,500,2000)
plt.legend()
```

```
plt.show()
print("Final R2 Score:",r2_score(y_2006,rf.predict(X_2006)))
```

```
R2 Score on test: 0.7109454039566507
```




```
Final R2 Score: 0.8857925371269807
```

03. Time Series Forecast

The previous models can predcit the energy of the hour based on some variables, however, we can see that both years have a similar pattern in the energy, but it has a different scale.

Hence, if we apply a tiem series forecast we might get a better model to predict the energy based on time

In [7]:
```
df_energy= pd.read_csv("/content/energy__data-2 (1).csv")
df_energy
```

Out[7]:

	Hour	energy_consumpt_2005	energy_consumpt_2006	full_temp_2005	full_humid_2005
0	1	631.623161	1246.300847	-0.400000	64.000000
1	2	534.397104	1062.500558	-0.733333	65.333333
2	3	453.538785	884.586887	-1.066667	66.666667
3	4	400.699718	786.564121	-1.400000	68.000000
4	5	378.171092	742.669614	-1.666667	60.333333
...	...	...	...	...	...
8779	8780	950.369306	0.000000	3.333333	64.000000
8780	8781	880.138770	0.000000	2.666667	68.000000
8781	8782	792.754026	0.000000	2.000000	72.000000
8782	8783	740.446668	0.000000	1.333333	76.000000
8783	8784	706.176769	0.000000	0.666667	80.000000

8784 rows × 7 columns

03.1 Na & Outliers

We will deal with Na and outliers so that we can get a better representation of the time series

In [8]:

```
#We will check for Na and interpolate it to get a close prediction of what tha
df_energy.isna().sum()
```