Actividad

Poryecto Regresion

Table of Contents

Fecha:

```
fecha = datetime('now', 'Format', 'dd-MM-yyyy');
disp(['Fecha actualizada: ', char(fecha)])
```

```
Fecha actualizada: 06-07-2024
```

Objetivos:

- **Desde Open Energy Data Initiative, seleccionar un archivo CSV cualquiera de cualquier estado:** https://openei.org/datasets/files/961/pub/COMMERCIAL_LOAD_DATA_E_PLUS_OUTPUT/
- Definir la variable de salida y las variables de entrada del sistema
- **Definir basado en alguna aplicacion,** Cambiar la resolucion temporal de los datos: n Dias **(Ajustar nhoras)**
- Seleccionar Variables o caracteristicas empleando la matriz de correlacion **(Ajustar el threshold)**
- **Seleccionar el algortimo de ML con un menos error de prediccion empleando el toolbox de Matlab Regression Learner**

- Dividir el dataset en 70% para entrenar y 30% validar **(Ajustar el PSplit)**
- Usando el algoritmo de ML se entrena el modelo de regression
- Cargar y validar el modelo entrenado
- Graficar el valor predecido vs el valor real
- **Analizar el resultado obtenido y tratar de reducir el error**

Nombre:

- sunombre

Repository:

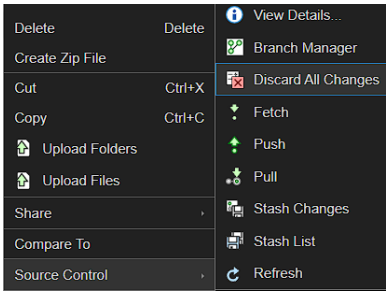
- https://github.com/vasanza/SSE

Librarys:

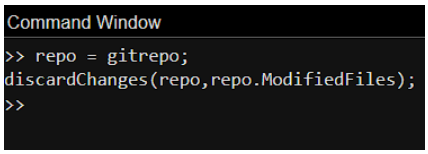
- https://github.com/vasanza/Matlab_Code
- https://es.mathworks.com/help/matlab/ref/ls.htm
- https://es.mathworks.com/help/matlab/ref/matlab.git.gitrepository.discardchanges.html#d126e406558

Paso 0: Descartar cualquier cambio realizado en el repositorio clonado

```
%Version Online, Opcion 1:
% Source Control -> Discard all changes
% Source Control -> git pull
```

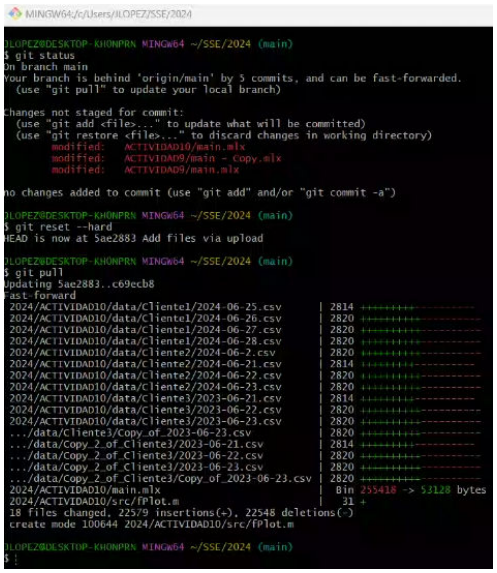



```
%Version Online, Opcion 2:
% repo = gitrepo;
% discardChanges(repo,repo.ModifiedFiles);
% Source Control -> git pull
```




```
% Version para PC, en el Bash del Git:
% git status
```

```
% git reset --hard
% Git pull
```



Paso 1: Limpiar variables y linea de comandos

```
clear % Para borrar el workspace y liberar memoria RAM
clc  % Limpiar el command window
```

Paso 2.- Configuración de carpeta ./src para librerias

```
%nombre de la carpeta donde estan los codigos
addpath(genpath('./src'));
```

Paso 3- Configuranción de carpeta de ./data para datasets

```
%Nombre de la carpeta donde estan los archvios csv
datapath=fullfile('./data/');
```

Paso 4- Desde Open Energy Data Initiative, seleccionar un archivo CSV cualquiera de cualquier estado

Informacion del dataset

- USA_AK_FAIRBANKS.csv
- USA_AK_Anchorage.Intl.AP.702730_TMY3/

```
cd data %Comando linux para entrar una carpeta
httpsUrl = "https://openei.org/datasets/files/961/pub/COMMERCIAL_LOAD_DATA_E_PLUS_OUTPUT/USA_AK
dataUrl = strcat(httpsUrl,"/RefBldgSmallHotelNew2004_v1.3_7.1_8A_USA_AK_FAIRBANKS.csv");
DataFile = "RefBldgSmallHotelNew2004_v1.3_7.1_8A_USA_AK_FAIRBANKS.csv";
DataFileFullPath = websave(DataFile,dataUrl);
```

```
cd .. %Comando linux para salir de la carpeta
clear httpsUrl dataUrl DataFile DataFileFullPath
```

Paso 4- Buscar los nombres y Cargar los datos de todos los archivos dentro de la carpeta ./data

```
%Leer un archivo csv y lo carga como una tabla
filename = FindCSV(datapath);
maxnames=size(filename,1);
% Dataset es una tabla donde cada columna es una variable con su
% respectivo nombre
index=1; % El archivo que quiero que lea desde la carpeta data
filename(index).name
```

```
ans =
'RefBldgSmallHotelNew2004_v1.3_7.1_8A_USA_AK_FAIRBANKS.csv'
```

```
Dataset=fLoadTableCSV_index(filename,datapath,index)
```

```
Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating
variable names for the table. The original column headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.
Dataset = 8760×11 table
```

...

	Date_Time	Electricity_Facility_kW__Hourly_	Fans_Electricity_kW__Hourly_
1	'01/01 01:0...	49.7663	4.9317
2	'01/01 02:0...	49.0528	4.9225
3	'01/01 03:0...	45.6644	4.9503
4	'01/01 04:0...	46.4704	4.9665
5	'01/01 05:0...	48.4044	4.9731
6	'01/01 06:0...	52.7617	4.9778
7	'01/01 07:0...	70.7867	4.9323
8	'01/01 08:0...	83.3511	4.8494
9	'01/01 09:0...	103.3132	4.7682
10	'01/01 10:0...	92.3172	4.8234
11	'01/01 11:0...	67.0967	4.9117
12	'01/01 12:0...	63.2558	4.9205
13	'01/01 13:0...	63.0341	4.9156
14	'01/01 14:0...	62.7386	4.9091

⋮

Paso 5- Extraer nombres de variables y crear datetime

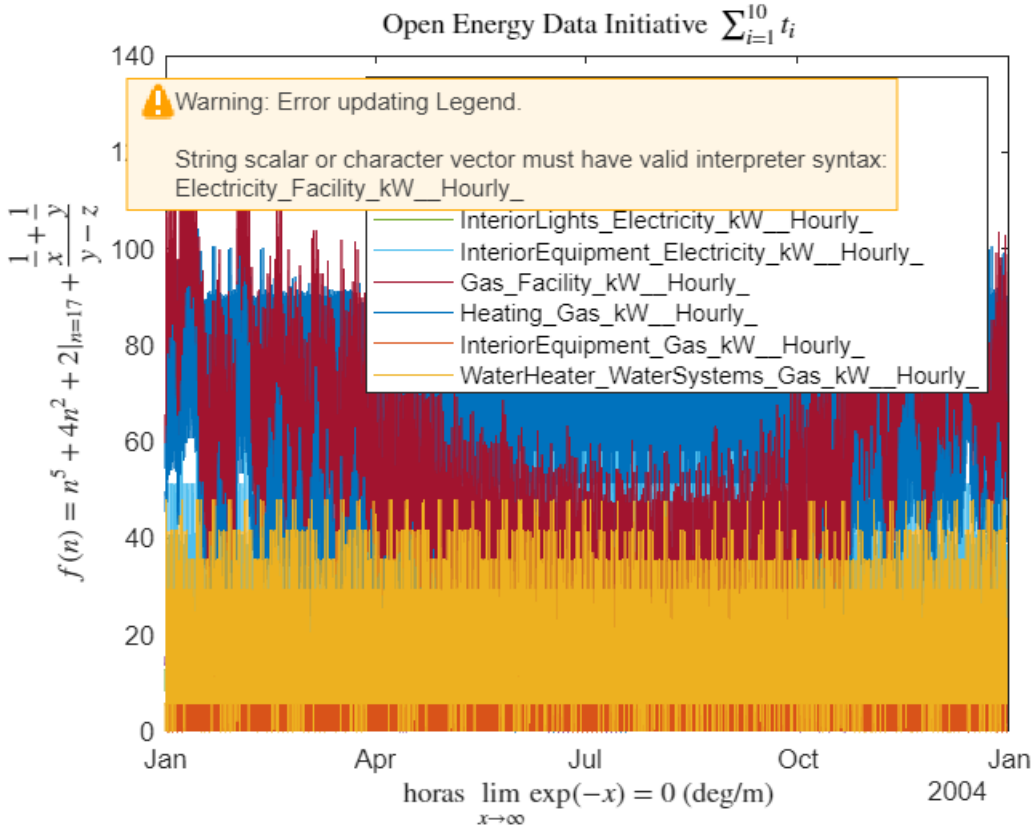
```
% Extraer todos los nombres de variables de la tabla
% Se hace el cast de cell a string
```

```
varnames=string(Dataset.Properties.VariableNames);
% Eliminar el primero nombbre de variable
varnames=varnames(2:end)';

% Esto es para eliminar el warning de los legend en el plot
%LegendNames=char(varnames);
%LegendNames=LegendNames(:,1:15);
%LegendNames=[LegendNames char(65*ones([size(varnames,1),1]))];
%LegendNames=string(LegendNames);

% Crear datatime con una frecuencia de muestreo de un dato por hora segun el
% dataset
%Time = Start Time: Step Time: End Time
time = datetime(2004, 1, 1):hours(1):datetime(2004, 12, 31);
% Se elimina el primer valor
time=time(1,2:end)';

% para agregar una nueva variables en la tabla
%Dataset.('Time_Stamp')=time;
```

Paso 6- Graficar todas las variables

```
figure;
% Dataset
Variable1=Dataset.(varnames(1));
plot(time,Variable1)
hold on
Variable2=Dataset.(varnames(2));
plot(time,Variable2)
Variable3=Dataset.(varnames(3));
plot(time,Variable3)
Variable4=Dataset.(varnames(4));
plot(time,Variable4)
Variable5=Dataset.(varnames(5));
plot(time,Variable5)
Variable6=Dataset.(varnames(6));
plot(time,Variable6)
Variable7=Dataset.(varnames(7));
plot(time,Variable7)
Variable8=Dataset.(varnames(8));
plot(time,Variable8)
Variable9=Dataset.(varnames(9));
plot(time,Variable9)
Variable10=Dataset.(varnames(10));
plot(time,Variable10)

hold off
%legend(LegendNames);
legend(varnames);
```

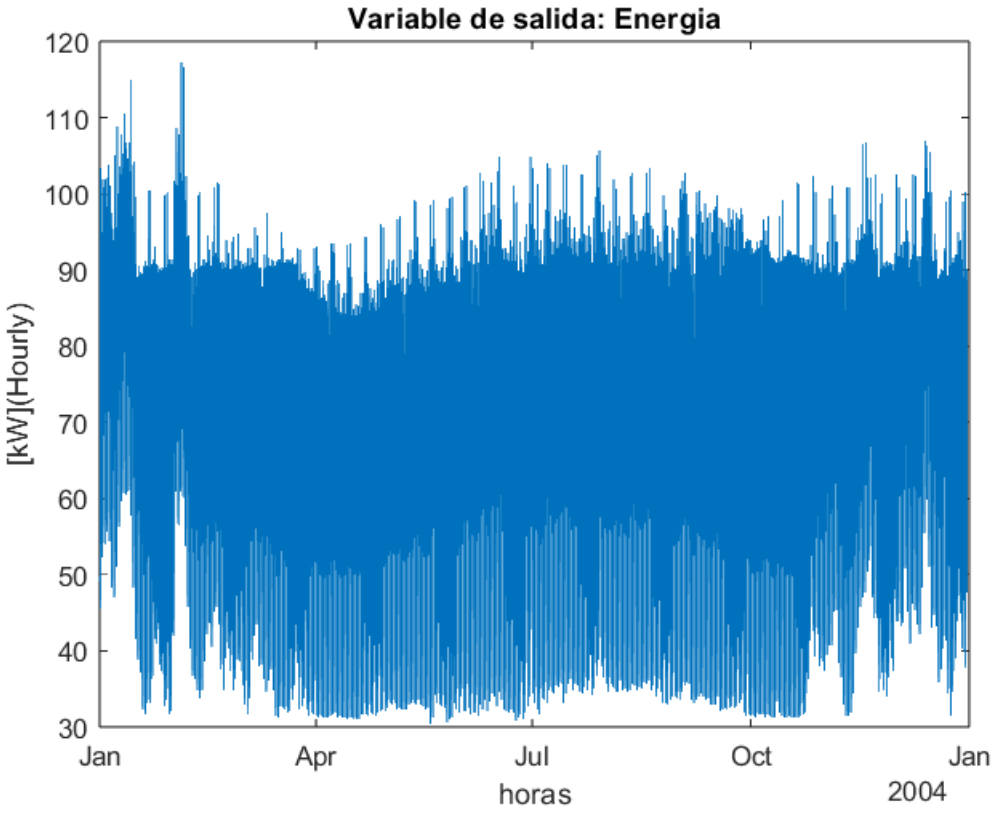
```
% tambien se puede usar latex para los lables y titles
% https://en.wikibooks.org/wiki/LaTeX/Mathematics
title('Open Energy Data Initiative $ \sum_{i=1}^{10} t_i $', 'interpreter', 'latex');
xlabel('horas $ \lim\limits_{x \to \infty} \exp(-x) = 0 $ (deg/m)', 'interpreter', 'latex')
ylabel(' $ f(n) = n^5 + 4n^2 + 2 |_{n=17} + \frac{\frac{1}{x}+\frac{1}{y}}{y-z} $', 'interprete
```




```
clear filename maxnames index
clear Variable1 Variable2 Variable3 Variable4 Variable5 Variable6 ...
    Variable7 Variable8 Variable9 Variable10
```

Paso 7- Definir la variable de salida y las variables de entrada del sistema

```
figure;
% Dataset
output=Dataset.(varnames(1));
plot(time,output)
title('Variable de salida: Energia');
xlabel('horas')
ylabel('[kW](Hourly)')
```

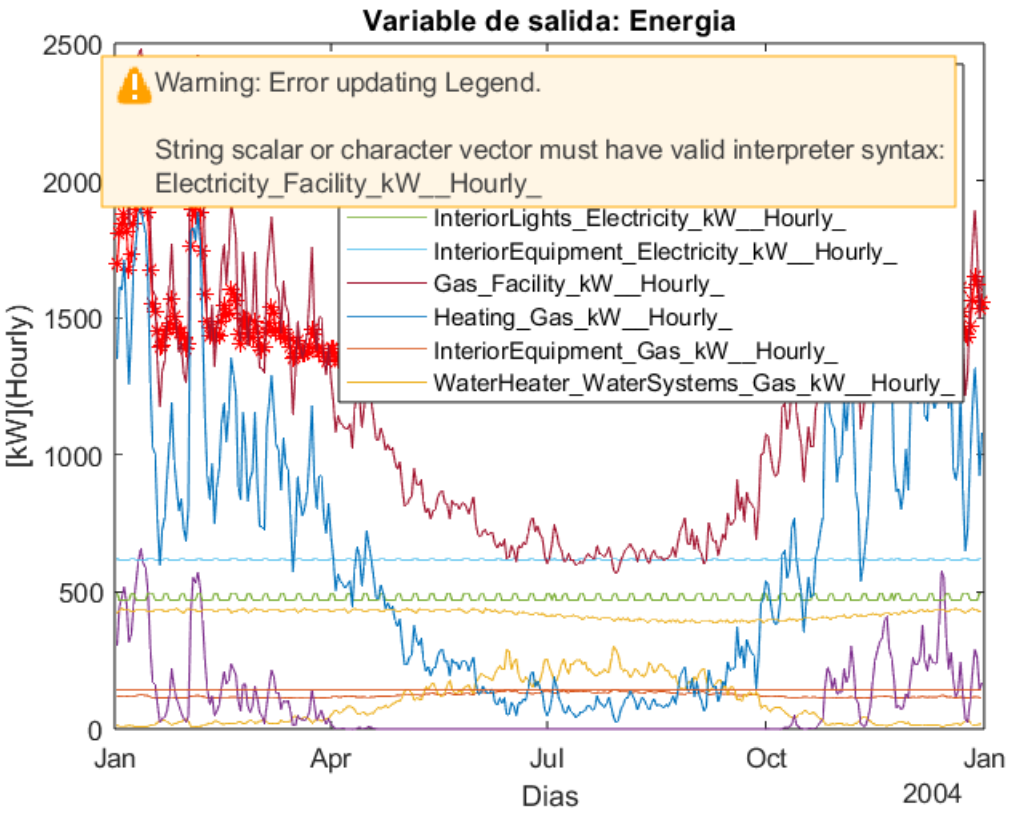



```
% Estamos usando las variables como caracteristicas
input=[Dataset.(varnames(2)) Dataset.(varnames(3))...
    Dataset.(varnames(4)) Dataset.(varnames(5))...
    Dataset.(varnames(6)) Dataset.(varnames(7))...
    Dataset.(varnames(8)) Dataset.(varnames(9))...
    Dataset.(varnames(10))];
```

Paso 8- Cambiar la resolucion temporal de los datos: Dias, Semanas, meses

```
%Variable de conversion de horas a Dias, Semanas o meses
nhoras=24*1; %dias

outputDias=[];
inputDias=[];
for i=1:nhoras:size(output,1)-(nhoras-1)
    outputDias=[outputDias;sum(output(i:i+nhoras-1),1)];%1+23 =24
    inputDias=[inputDias;sum(input(i:i+nhoras-1,:),1)];%1+23 =24
end

% Crear datatime con una frecuencia de muestreo de un dato por hora segun el
% dataset
%Time = Start Time: Step Time: End Time
timeDias = datetime(2004, 1, 1):hours(nhoras):datetime(2004, 12, 31);
% Se elimina el primer valor
timeDias=timeDias(1,2:end)';
```

```
figure;
plot(timeDias,outputDias,'-*r')
hold on
plot(timeDias,inputDias)
hold off
title('Variable de salida: Energia');
xlabel('Dias')
ylabel('[kW](Hourly)')
legend(varnames);
```

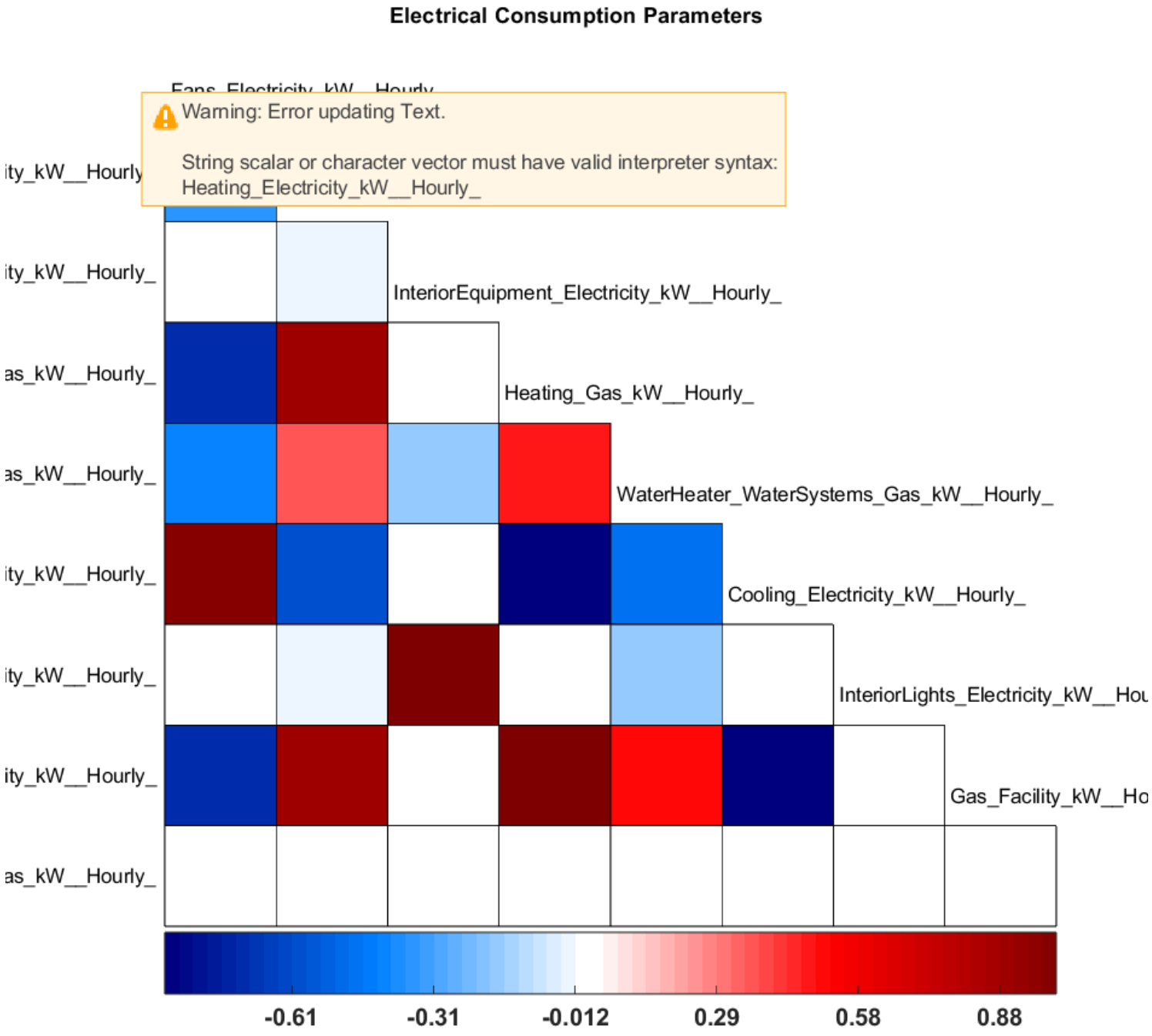



```
clear i nhoras;
```

Paso 9- Seleccionar Variables o caracteristicas

```
%Maximum correlation value allowed
% Dafault 0.75
threshold = 0.6;

% El numero de variables a analizar debe ser igual al numero de nombres de
% variables
Features_labels=cellstr(varnames(2:end)');
%corrcoef(input)

% Example:
% a=[1:10];b=a+3;c=a.*b;
% corrcoef([a b c])
```

```
% 1.0000    1.0000    0.9816
% 1.0000    1.0000    0.9816
% 0.9816    0.9816    1.0000

[NewDataFeatures,NewFeaturesLabels,LabelsRemove] = Feature_Selection(inputDias,Features_labels,
```




```
NewDataFeatures = 365×5
  117.5758  305.9132  496.0802  144.2000  422.7136
  120.2368  446.2676  470.2083  144.2000  433.6379
  121.0207  480.2288  470.2083  144.2000  439.3978
  121.7134  517.9401  470.2083  144.2000  433.6287
  120.6447  455.0323  470.2083  144.2000  433.5425
  118.0155  316.2266  470.2083  144.2000  433.5434
  118.5121  348.8277  495.8694  144.2000  428.3304
  120.4338  454.6374  496.0802  144.2000  428.4893
  122.0624  538.2837  470.2083  144.2000  433.5442
```

```
  123.5053   612.5234   470.2083   144.2000   433.5407
      ⋮
NewFeaturesLabels = 1×5 cell
'Fans_Electricity_kW__Hourly_''Heating_Electricity_kW__Hourly_''InteriorLights_E ···
LabelsRemove = 1×4 cell
'Cooling_Electricity_kW__Hourly_''InteriorEquipment_Electricity_kW__Hourly_''Gas ···
```

```
%Variables eliminadas por superarf el threshold
LabelsRemove'
```

```
ans = 4×1 cell
'Cooling_Electricity_kW__Hourly_'
'InteriorEquipment_Electricity_kW__Hourly_'
'Gas_Facility_kW__Hourly_'
'Heating_Gas_kW__Hourly_'
```

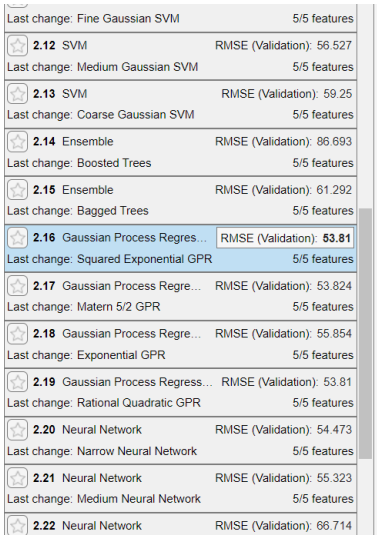
```
%variables que se quedan por no superan el threshold en coreelacion
NewFeaturesLabels'
```

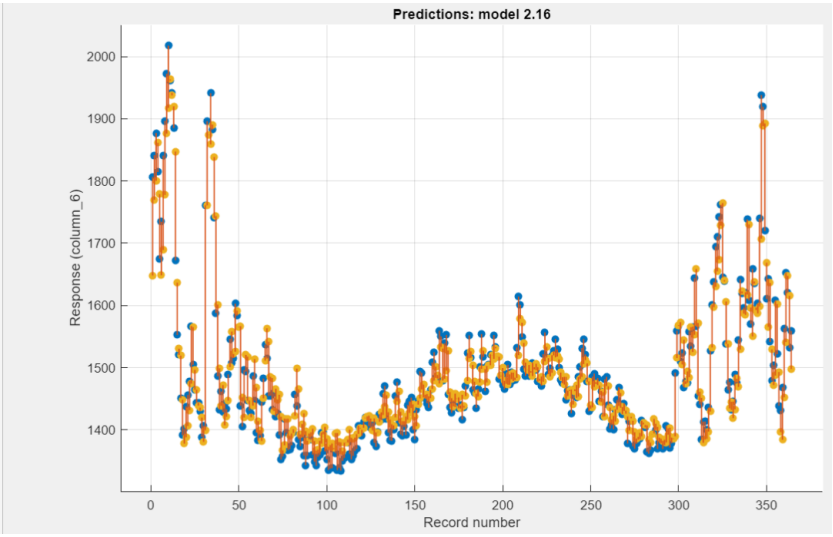
```
ans = 5×1 cell
'Fans_Electricity_kW__Hourly_'
'Heating_Electricity_kW__Hourly_'
'InteriorLights_Electricity_kW__Hourly_'
'InteriorEquipment_Gas_kW__Hourly_'
'WaterHeater_WaterSystems_Gas_kW__Hourly_'
```

```
clear threshold Features_labels LabelsRemove;
```

Paso 10- Seleccionar el algortimo de ML con un menos error de prediccion empleando el toolbox de Matlab Regression Learner

```
% Concatenando la variables de entrada actual con
% la salida al dia siguiente
DataRegression=[NewDataFeatures(1:end-1,:) outputDias(2:end,1)];
regressionLearner
```





Model 2.16: Gaussian Process Regression
Status: Trained

Training Results

RMSE (Validation)	53.81
R-Squared (Validation)	0.81
MSE (Validation)	2895.5
MAE (Validation)	37.747
Prediction speed	~5800 obs/sec
Training time	19.968 sec

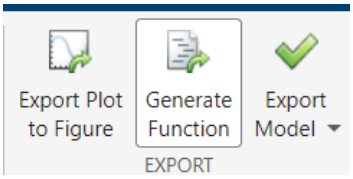
Model Hyperparameters

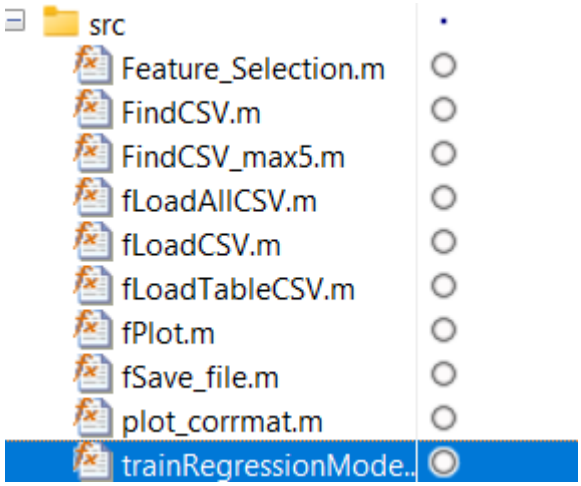
Preset: Squared Exponential GPR
Basis function: Constant
Kernel function: Squared Exponential
Use isotropic kernel: true
Kernel scale: Automatic
Signal standard deviation: Automatic
Sigma: Automatic
Standardize: true
Optimize numeric parameters: true

Feature Selection: 5/5 individual features selected

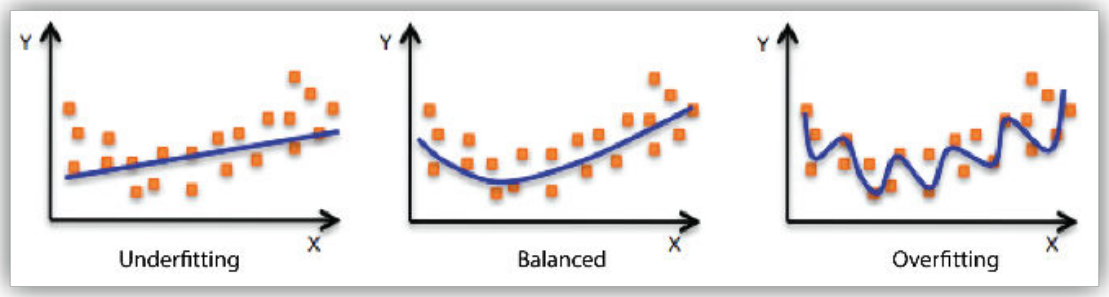
PCA: Disabled

Optimizer: Not applicable



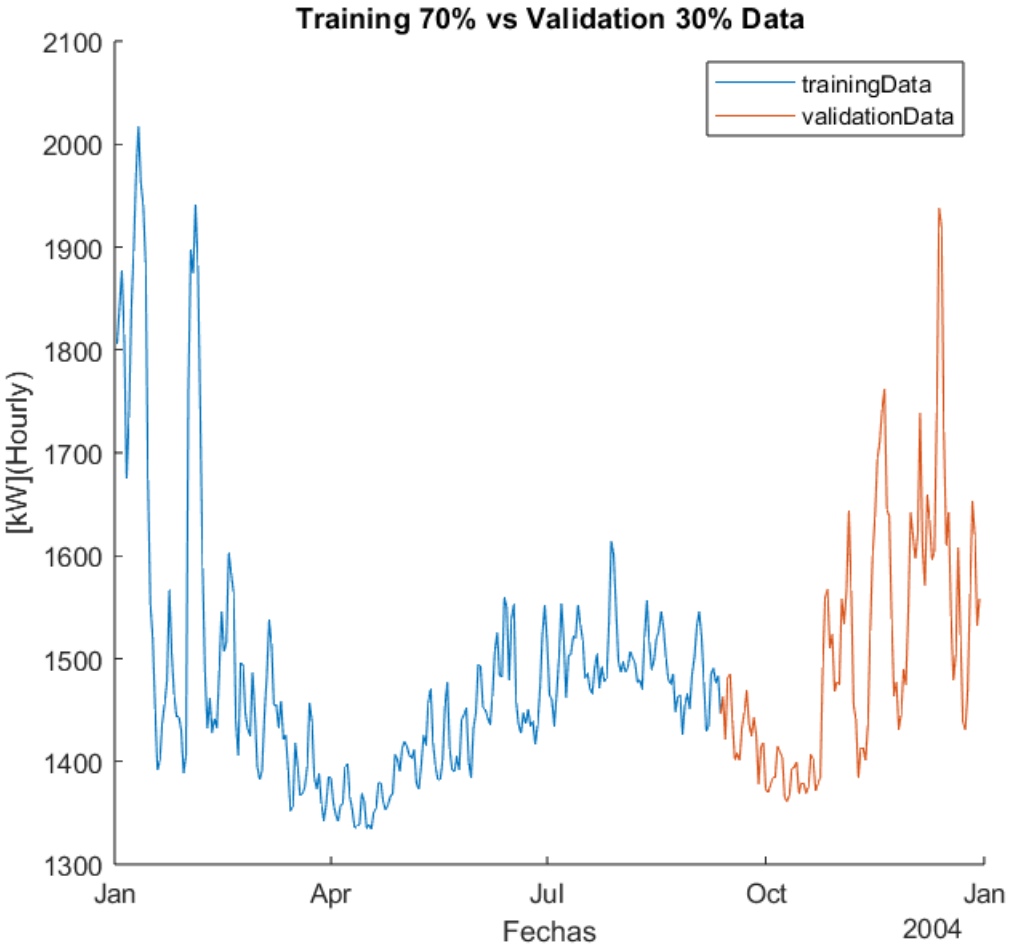


Paso 11- Dividir el dataset en 70% para entrenar y 30% validar




```
%Set de % de entrenamiento
% Dafault 0.70
PSplit = 0.7;
```

```
% Dataset de entrenamiento 70%
trainingData=DataRegression(1:round(end*PSplit),:);
traninTime=timeDias(1:round(end*PSplit));
%Calcular el minimo numero de filas entre Data y Time
nfilas=min([size(trainingData,1) size(traninTime,1)]);
%Tomamos el valor minimo de nfilas como maixmo de filas
trainingData=trainingData(1:nfilas,:);
traninTime=timeDias(1:nfilas,:);

% Dataset de validcion 30%
validationData=DataRegression(round(end*PSplit):end,:);
validationTime=timeDias(round(end*PSplit):end);
%Calcular el minimo numero de filas entre Data y Time
nfilas=min([size(validationData,1) size(validationTime,1)]);
%Tomamos el valor minimo de nfilas como maixmo de filas
validationData=validationData(1:nfilas,:);
validationTime=validationTime(1:nfilas,:);

figure;
% Dataset
hold on
plot(traninTime,trainingData(:,end))
plot(validationTime,validationData(:,end))
hold off
title(['Training ' num2str(PSplit*100) '% vs Validation '...
    num2str(100-PSplit*100) '% Data']);
xlabel('Fechas')
ylabel('[kW](Hourly)')
legend('trainingData','validationData')
```




```
clear PSplit;
```

Paso 12- Usando el algoritmo de ML se entrena el modelo de regression (costo computacional)

```
%Esta funcion permite generar un modelo actualizado cada vez que se ejecuta
% Siempre que el numero de variables de entrada sea la misma y la cantidad
% de nuevos datos no sea muy alta
[trainedModel, validationRMSE] = trainRegressionModel(trainingData);
```

```
Warning: Iteration limit reached.
Warning: Regression design matrix is rank deficient to within machine precision.
Warning: Regression design matrix is rank deficient to within machine precision.
Warning: Regression design matrix is rank deficient to within machine precision.
Warning: Regression design matrix is rank deficient to within machine precision.
Warning: Iteration limit reached.
Warning: Regression design matrix is rank deficient to within machine precision.
```

```
Warning: Regression design matrix is rank deficient to within machine precision.
```

```
%Este es el error de entrenaamiento
validationRMSE
```

```
validationRMSE = 49.9473
```

```
%Permite guardar el modelo entrenado que se encuentra en el workspace
save("trainedModel.mat","trainedModel")
```

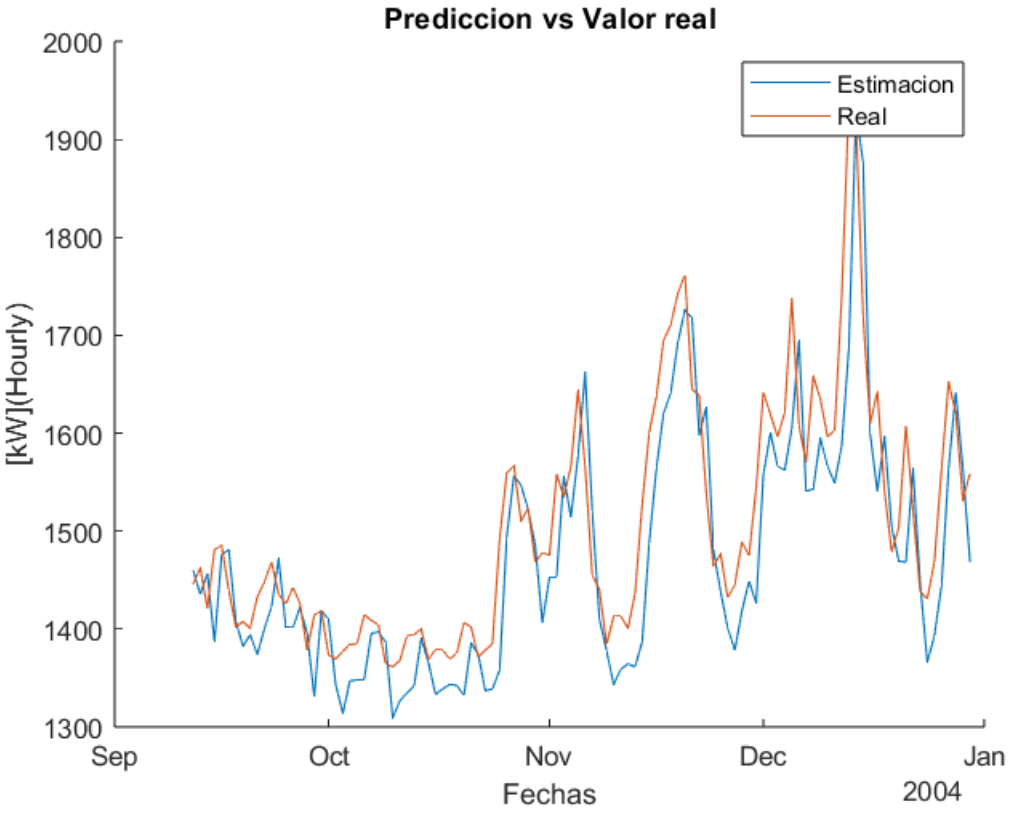
Paso 13- Cargar y validar el modelo entrenado

```
%Cargar el modelo entrenado y guardado
load("trainedModel.mat")

%Usar el modelo entrenado para predecir valores de consumo de energia
yest = trainedModel.predictFcn(validationData(:,1:end-1));

%El valor real de consumo de energia para comparar
yout = validationData(:,end);

%Error de prediccion con datos de validacion
validationRMSE = sqrt(mean((yest - yout).^2))
```

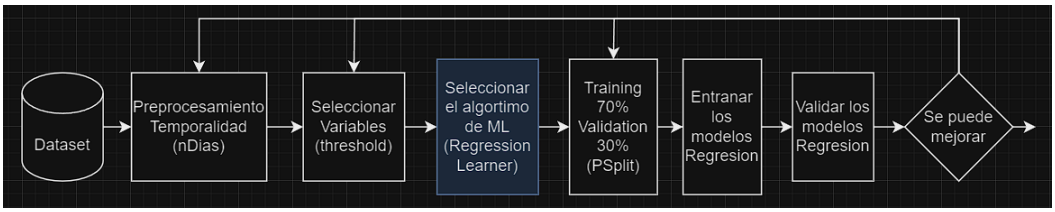
```
validationRMSE = 63.3533
```

paso 14 - Graficar el valor predecido vs el valor real

```
figure;
% Dataset
hold on
plot(validationTime,yest)
plot(validationTime,yout)
title('Prediccion vs Valor real');
xlabel('Fechas')
ylabel('[kW](Hourly)')
legend('Estimacion','Real')
```



paso 15 - Mejorar el modelo de prediccion



Con el dataset USA_AK_FAIRBANKS.csv se llegaron a obtener los siguientes Resultados:

- Para predicciones de meses, al tener solo 12, se vuelve necesario tener mas informacion por parte de las variables de entrada. Es decir, no importa la redundancia en las variables de entrada de la informacion por que ayudara a que el modelo tenga un menor error. En este ejemplo, se puso un threshold=0 y el eror de prediccion se vio decrementado.
- Para predicciones de dias, al tener mas dias y por tanto mas ejemplos, la redundancia de las variables de entrada se vuelve contraproducente. Es decir, al usar un threshold=0.75 se eliminaron 4 variables de entrada y esto mejoro la prediccion del modelo de regresion.