

Simulación de Sistemas Eléctricos

Actividad 8: Modelo de predicción con una Var.

- **Variable independiente (x):** Energía, Voltaje.....,etc. en $t=t_0$ (Valor actual)
- **Variable dependiente (f(x)):** Energía en $t=t_0+dt$ (Valor siguiente); $dt=1\text{seg}$
- **Frecuencia de muestreo de los datos:** 1 muestra por segundo (sps)
- **Venta temporal de predicción:** ($dt=1\text{día}$)

Nombres:

- Estudiante 1: VA
- Estudiante 2: VA

Contenido:

- Descargar datos (.csv) desde repositorio (DataPort)
- **Paper:** <https://doi.org/10.1016/j.procs.2022.07.035>
- **Dataset:** <https://ieee-dataport.org/open-access/data-server-energy-consumption-dataset>
- **Código Matlab:** <https://github.com/vasanza/EnergyConsumptionPrediction>
- **Funciones de Matlab:** https://github.com/vasanza/Matlab_Code

Preprocesamiento (*)

- 1.- Leer el archivo .csv usando Matlab y graficar los datos
- 2.- Cambiar la frecuencia de muestreo de los datos (1 muestra por día)
- 3.- Adecuar el dataset con la var. independiente (x) y la var. dependiente f(x)

Extracción de características (*)

- Pendiente

Selección de características

- Pendiente

Modelo de predicción

- 4.- Dataset de training (70%) y testing (30%)
- 5.- ToolBox de Matlab - Regression Learner (24 algoritmos)
- 6.- Seleccionar el modelo que tiene un menor error de predicción
- 7.- Exportar la función del mejor modelo
- 8.- Calcular el error de predicción usando el dataset de testing
- **Nota:** realizar cualquier cambio que usted considere adecuado
- (*) Despues:

- Usar la matriz de correlación para eliminar del dataset var. redundantes
- Filtrar datos (preprocesamiento)
- Incrementar las características temporales

Eliminar datos de memoria y command window

```
clear; %Borrar el kspace
clc; %Borrar el command window
```

Configuración de carpeta /src para librerías

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

Configuración de carpeta de /data para datasets

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

1.- Leer el archivo .csv usando Matlab y graficar los datos

```
%Esto es lo que se hace con datos .csv
data=readtable(fullfile(datapath,filenames(1).name));
%clear dataRaw; %Borrar de memoria la variable que no usare despues
%Seleccionar las variables electricas de interes
data2=table2array(data(:,1:6)); % Convertir datos de string a float o int
data2(isnan(data2))=0; % Eliminar los datos NaN
```

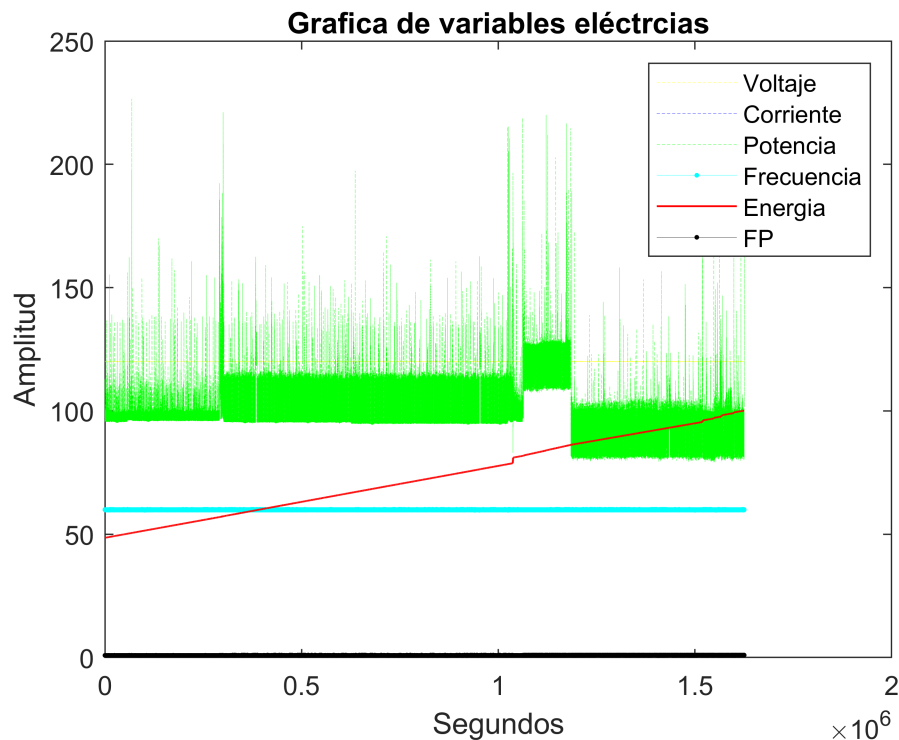
Graficar todas las Variables

```
Voltaje=data2(:,1);
Corriente=data2(:,2);
Potencia=data2(:,3);
Frecuencia=data2(:,4);
Energia=data2(:,5);
FP=data2(:,6);

figure %crear una nueva grafica
%plot(data2) %plotea la grafica 1
plot(Voltaje,'y--',LineWidth=0.1);
hold on %habilita plotear mas de una grafica en la misma figura
plot(Corriente,'b--',LineWidth=0.1);
plot(Potencia,'g--',LineWidth=0.01);
plot(Frecuencia,'c.-',LineWidth=0.1);
plot(Energia,'r',LineWidth=0.7);
plot(FP,'k.-',LineWidth=0.1);
```

```
hold off
```

```
title('Grafica de variables eléctricas');  
xlabel('Segundos');  
ylabel('Amplitud');  
legend('Voltaje','Corriente','Potencia','Frecuencia','Energia','FP');
```



```
%Limpiar del workspace variables no utilizadas  
clear data; clear datapath;
```

2.- Cambiar la frecuencia de muestreo de los datos (1 muestra por día)

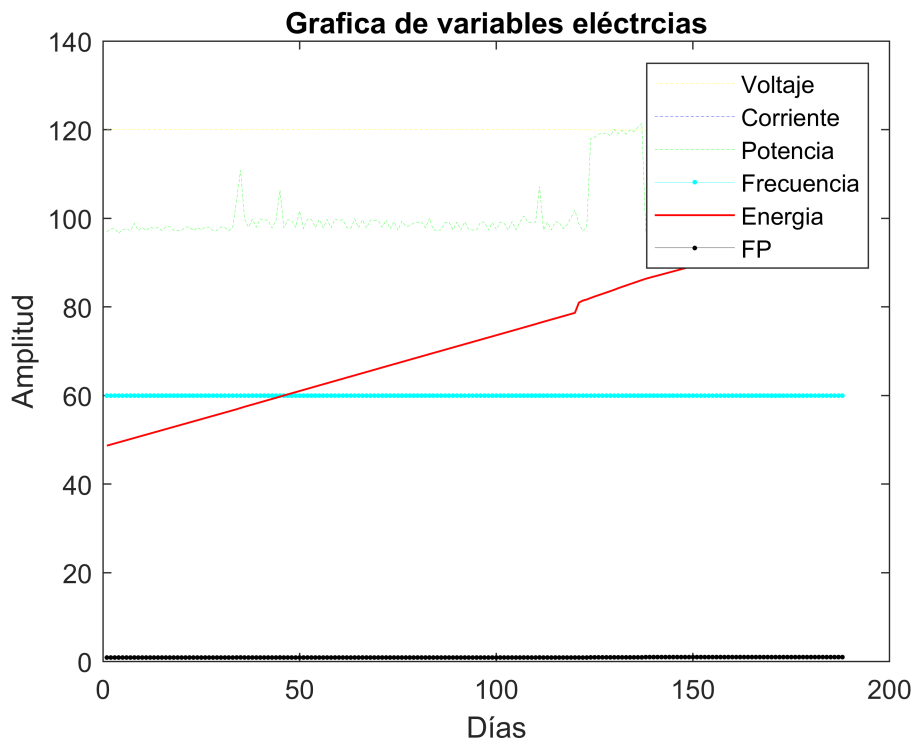
```
win=360*24; %dado a que los datos estan en segundos y queremos llevarlo a dias  
data2Mean=fData_MeanWin(data2,win);%60 segundos
```

```
figure %crear una nueva grafica  
plot(data2Mean(:,1),'y--',LineWidth=0.1);  
hold on %habilita plotear mas de una grafica en la misma figura  
plot(data2Mean(:,2),'b--',LineWidth=0.1);  
plot(data2Mean(:,3),'g--',LineWidth=0.01);  
plot(data2Mean(:,4),'c.-',LineWidth=0.1);  
plot(data2Mean(:,5),'r',LineWidth=0.7);  
plot(data2Mean(:,6),'k.-',LineWidth=0.1);  
hold off
```

```

title('Grafica de variables eléctricas');
xlabel('Días');
ylabel('Amplitud');
legend('Voltaje','Corriente','Potencia','Frecuencia','Energia','FP');

```



```

%Limpiar del workspace variables no utilizadas
clear data2; clear FP; clear Potencia; clear Frecuencia; clear Energia;
clear Corriente; clear Voltaje; clear win;

```

3.- Adecuador el dataset con la var. independiente (x) y la var. dependiente f(x)

Variables independiente (x): Energía, Voltaje.....,etc. en t=t0 (Valor actual)

```

Voltaje=data2Mean(:,1);
Corriente=data2Mean(:,2);
Potencia=data2Mean(:,3);
Frecuencia=data2Mean(:,4);
Energia=data2Mean(:,5);
FP=data2Mean(:,6);

% Variables independientes (x1,...,xn)
%input=[Voltaje(1:size(Voltaje,1)-1,1) Corriente(1:size(Corriente,1)-1,1)...
% Potencia(1:size(Potencia,1)-1,1) Frecuencia(1:size(Frecuencia,1)-1,1)...
% Energia(1:size(Energia,1)-1,1) FP(1:size(FP,1)-1,1)];
input=[Energia(1:size(Energia,1)-1,1)];
output=Energia(2:size(Energia,1),1);% Variable dependient f(x)

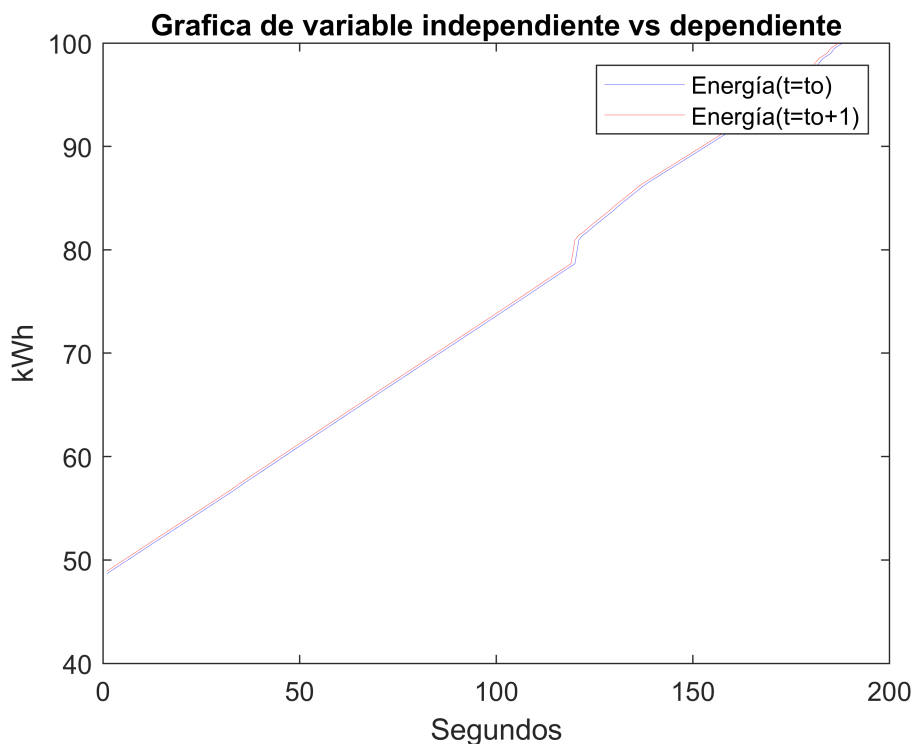
```

```

%Primeras columnas son las entradas o varianles independientes
%La ultima columna es la salida o variable dependiente
dataset=[input output];

figure %crear una nueva grafica
plot(Energia(:,1),'b',LineWidth=0.1);
hold on %habilita plotear mas de una grafica en la misma figura
plot(output(:,1),'r',LineWidth=0.1);
hold off
%
title('Grafica de variable independiente vs dependiente');
xlabel('Segundos');
ylabel('kWh');
legend('Energía(t=to)', 'Energía(t=to+1)');

```



```

%Limpiar del workspace variables no utilizadas
clear FP; clear Potencia; clear Frecuencia; clear Energia;
clear Corriente; clear Voltaje; clear data2Mean;
clear input; clear output;

```

4.- Dataset de training (70%) y testing (30%)

```

%70 de entrenamiento y 30% de testing
Training=dataset(1:round(size(dataset,1)*0.7),:);% fila 1 hasta la fila del 70% de todas las f
%desde la fila del 70% de todas las filas hasta la ultima fila
Testing=dataset(round(size(dataset,1)*0.7)+1:round(size(dataset,1)),:);

```

5.- ToolBox de Matlab - Regression Learner (24 algoritmos)

regressionLearner

Data set

Data Set Variable

dataset 1625174x2 double ▼

☒ Use columns as variables
☐ Use rows as variables

Response

☒ From data set variable
☐ From workspace

column_2 double 48.56 ... ▼

Predictors

	Name	Type	Range
<input checked="" type="checkbox"/>	column_1	double	48.56 .. 100.12
<input type="checkbox"/>	column_2	double	48.56 .. 100.12

[How to prepare data](#) [Refresh](#)

[Add All](#) [Remove All](#)

Validation

Validation Scheme

Cross-Validation ▼

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds: 10 ▲ ▼

[Read about validation](#)

Test

☐ Set aside a test data set

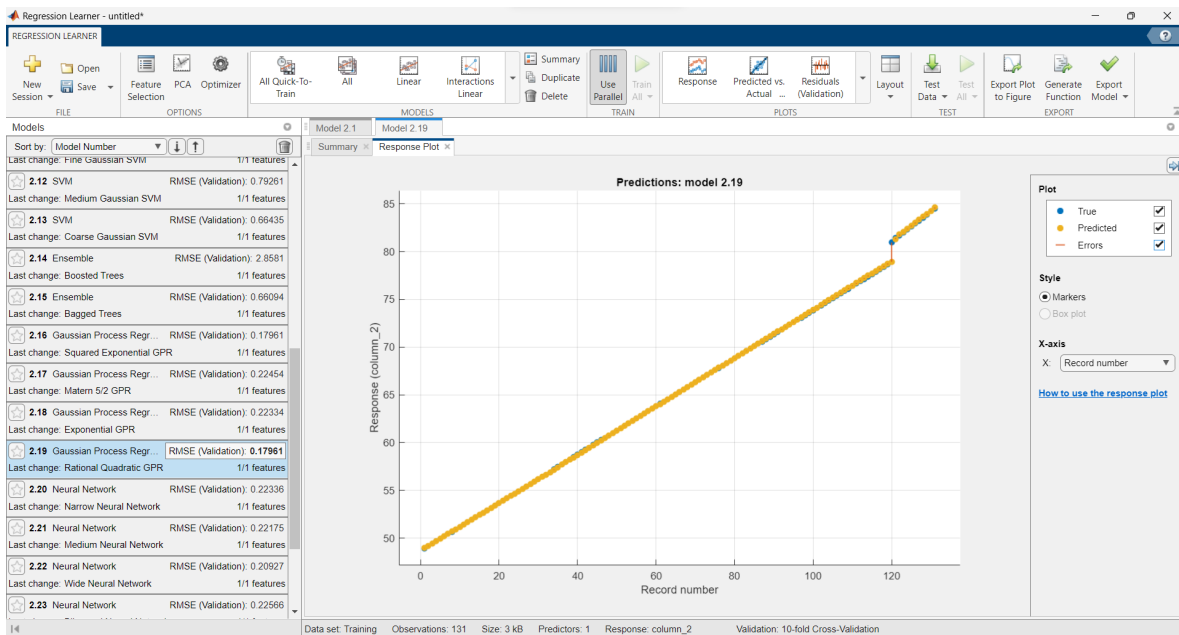
Percent set aside: 0 ▲ ▼

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

[Read about test data](#)

[Start Session](#) [Cancel](#)

6.- Seleccionar el modelo que tiene un menor error de predicción



7.- Exportar la función del mejor modelo

Model 2.19: Gaussian Process Regression
Status: Trained

Training Results

RMSE (Validation) 0.17961
R-Squared (Validation) 1.00
MSE (Validation) 0.032258
MAE (Validation) 0.040479
Prediction speed ~3100 obs/sec
Training time 5.5701 sec

Model Hyperparameters

Feature Selection: 1/1 individual features selected

PCA: Disabled

Optimizer: Not applicable

%Gaussian Process Regression

```
[trainedModel, validationRMSE] = trainRegressionModel(Training)
```

trainedModel = struct with fields:

predictFcn: @(x)gpPredictFcn(predictorExtractionFcn(x))

RegressionGP: [1x1 RegressionGP]

About: 'This struct is a trained model exported from Regression Learner R2022a.'

HowToPredict: 'To make predictions on a new predictor column matrix, X, use: ↳ yfit = c.predictFcn(X) ↳replac

validationRMSE = 0.1797

%Aquí le damos al modelo solo las variables de entrada

```
EnergiaEst=trainedModel.predictFcn(Testing(:,1));
```

figure %crear una nueva grafica

```
plot(Testing(:,2),'b',LineWidth=0.1);
```

hold on %habilita plotear mas de una grafica en la misma figura

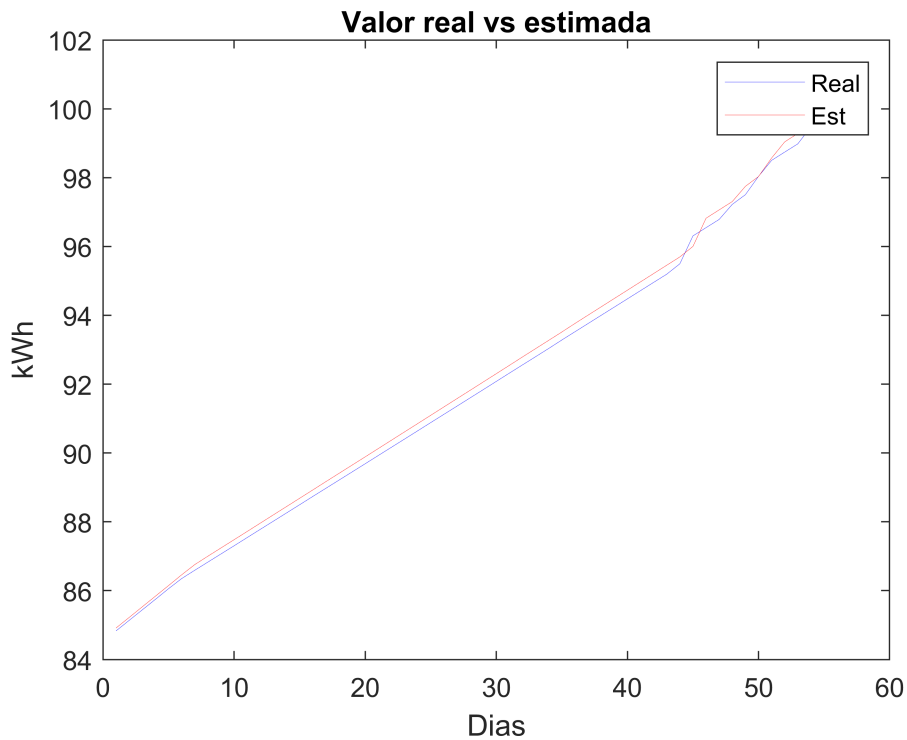
```
plot(EnergiaEst,'r',LineWidth=0.1);
```

hold off

%

```
title('Valor real vs estimada');
```

```
xlabel('Dias');
ylabel('kWh');
legend('Real', 'Est');
```



8.- Calcular el error de predicción usando el dataset de testing

```
%[rmse,mse,mae] = fBar_RmseMseMae(EnergiaEst,Testing(:,2))
rmse = sqrt(immse(EnergiaEst,Testing(:,2)))
```

```
rmse = 0.2103
```

```
mse = immse(EnergiaEst,Testing(:,2))
```

```
mse = 0.0442
```

```
mae = sum(abs(EnergiaEst-Testing(:,2))/length(EnergiaEst))
```

```
mae = 0.1988
```