

### 3.7. Regresión Múltiple con Validación

#### Ejercicio 102\_01. Regresión múltiple con validación en Python

```
import pandas as pd
from sklearn.model_selection import KFold
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

# Cargar los datos de entrenamiento
data_entrenamiento = pd.read_excel('tu_archivo_entrenamiento.xlsx')
X = data_entrenamiento.iloc[:, :-1] # Características
y = data_entrenamiento.iloc[:, -1] # Variable objetivo

# Definir el número de folds para la validación cruzada
k = 5

# Crear un objeto KFold
kf = KFold(n_splits=k, shuffle=True, random_state=42)

# Inicializar listas para almacenar las métricas
mse_scores = []
r2_scores = []
mae_scores = []

# Iterar sobre cada fold
for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Entrenar el modelo
    model = LinearRegression()
    model.fit(X_train, y_train)


    # Hacer predicciones
    y_pred = model.predict(X_test)

    # Calcular las métricas
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)

    mse_scores.append(mse)
    r2_scores.append(r2)
    mae_scores.append(mae)

# Calcular los promedios de las métricas
mse_mean = np.mean(mse_scores)
r2_mean = np.mean(r2_scores)
mae_mean = np.mean(mae_scores)

print("Error cuadrático medio promedio:", mse_mean)
print("Coeficiente de determinación promedio:", r2_mean)
```

**Autores:** María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
print("Error absoluto medio promedio:", mae_mean)
```

### Ejercicio 102\_02. Regresión múltiple con validación en MATLAB

```
function [mse_mean, r2_mean, mae_mean] = regresion_lineal_cv(X, y, k)
% REGRESION_LINEAL_CV Realiza regresión lineal múltiple con validación cruzada k-
fold
%
% Entrada:
%   X: Matriz de características (cada fila es una observación, cada columna
una característica)
%   y: Vector de valores objetivo
%   k: Número de folds para la validación cruzada
%
% Salida:
%   mse_mean: Error cuadrático medio promedio
%   r2_mean: Coeficiente de determinación promedio
%   mae_mean: Error absoluto medio promedio

% Crear un objeto de partición k-fold
cv = cvpartition(size(X,1),'Kfold',k);

% Inicializar vectores para almacenar las métricas
mse_scores = zeros(k,1);
r2_scores = zeros(k,1);
mae_scores = zeros(k,1);


% Iterar sobre cada fold
for i = 1:k
    % Obtener índices de entrenamiento y prueba
    idx = cv.test(i);
    X_train = X(~idx,:);
    y_train = y(~idx);
    X_test = X(idx,:);
    y_test = y(idx);

    % Entrenar el modelo
    mdl = fitlm(X_train, y_train);

    % Hacer predicciones
    y_pred = predict(mdl, X_test);

    % Calcular métricas
    mse_scores(i) = mean((y_test - y_pred).^2);
    r2_scores(i) = 1 - sum((y_test - y_pred).^2) / sum((y_test -
mean(y_test)).^2);
    mae_scores(i) = mean(abs(y_test - y_pred));
end

% Calcular los promedios de las métricas
mse_mean = mean(mse_scores);
r2_mean = mean(r2_scores);
mae_mean = mean(mae_scores);
```

**Autores:** María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

end

#### Cómo usar la función:

1. **Cargar los datos:** Carga tus datos en matrices X (características) e y (variable objetivo).
2. **Definir el número de folds:** Especifica el valor de k.
3. **Llamar a la función:** Llama a la función `regresion_lineal_cv` con tus datos y el valor de k.
4. **Interpretar los resultados:** La función devolverá el error cuadrático medio promedio, el coeficiente de determinación promedio y el error absoluto medio promedio.

```
% Cargar datos (suponiendo que ya están cargados en X e y)
```

```
k = 5;
```

```
% Llamar a la función
```

```
[mse_mean, r2_mean, mae_mean] = regresion_lineal_cv(X, y, k);
```

```
% Mostrar resultados
```

```
fprintf('Error cuadrático medio promedio: %.4f\n', mse_mean);
```

```
fprintf('Coeficiente de determinación promedio: %.4f\n', r2_mean);
```

```
fprintf('Error absoluto medio promedio: %.4f\n', mae_mean);
```