

3.6. Regresión Múltiple

Ejercicio 100_01. Regresión múltiple sin validación en Python:

```
filename = 'datos.xlsx'
data = pd.read_excel(filename)

# Asumimos que la última columna es la variable dependiente y
# y el resto son las variables independientes x
x = data.iloc[:, :-1] # Variables independientes (todas menos la última columna)
y = data.iloc[:, -1] # Variable dependiente (última columna)

# Crear el modelo de regresión lineal
modelo = LinearRegression()

# Ajustar el modelo con los datos
modelo.fit(x, y)

# Predecir los valores de y
y_est = modelo.predict(x)

# Mostrar los coeficientes y el intercepto
print("Coeficientes de la regresión:", modelo.coef_)
print("Intercepto:", modelo.intercept_)

# Calcular el coeficiente de determinación (R^2)
r2 = modelo.score(x, y)
print(f"R-cuadrado: {r2}")

# Visualizar los valores reales vs estimados
plt.scatter(y, y_est)
plt.xlabel('Valores reales de y')
plt.ylabel('Valores estimados de y')
plt.title('Comparación de valores reales vs estimados')
plt.grid(True)
plt.show()
```

Descripción del código:

- Lectura de datos:** Se carga el archivo Excel y se guarda en una tabla data: Usamos **pandas** para leer el archivo Excel. Aquí, el archivo se llama datos.xlsx (puedes reemplazarlo por tu archivo). El contenido se carga en un dataframe llamado data.**data:** El contenido del archivo Excel se guarda en la variable data como una tabla de datos. Las columnas representan las variables independientes (x) y dependiente (y).
- Separación de variables:** La última columna es la variable dependiente (y), y el resto son las variables independientes (x).
- Regresión lineal múltiple:** Creamos un modelo de regresión lineal múltiple con la función de **scikit-learn**.
- Resultados:** El modelo ajustado se muestra en pantalla, y se calculan los valores estimados de y.
- Visualización:** Un gráfico de dispersión compara los valores reales y estimados de y.

6. **Estadísticas:** Se muestra el coeficiente de determinación R^2 , que indica la calidad del ajuste.

Ejercicio 100_02. Regresión múltiple sin validación en MATLAB

```
% Cargar los datos desde el archivo Excel
% Reemplaza 'datos.xlsx' con el nombre de tu archivo Excel
filename = 'datos.xlsx';
data = readtable(filename);

% Asumimos que la última columna es la variable dependiente y
% y el resto son las variables independientes x
x = data{:, 1:end-1}; % Variables independientes
y = data{:, end};     % Variable dependiente

% Realizar la regresión lineal múltiple
mdl = fitlm(x, y);

% Mostrar los resultados de la regresión
disp(mdl);


% Predecir valores de y utilizando el modelo ajustado
y_est = predict(mdl, x);

% Visualizar los valores reales vs estimados
figure;
scatter(y, y_est);
xlabel('Valores reales de y');
ylabel('Valores estimados de y');
title('Comparación de valores reales vs estimados');
grid on;

% Mostrar estadísticas clave de la regresión
disp('R-cuadrado:');
disp(mdl.Rsquared.Ordinary);
```

Descripción del código:

1. **Lectura de datos:** Se carga el archivo Excel y se guarda en una tabla data:
 - **readtable:** Esta función carga los datos desde el archivo Excel. El archivo se llama datos.xlsx (puedes cambiar el nombre por el de tu archivo).
 - **data:** El contenido del archivo Excel se guarda en la variable data como una tabla de datos. Las columnas representan las variables independientes (x) y dependiente (y).
2. **Separación de variables:** La última columna es la variable dependiente (y), y el resto son las variables independientes (x).
3. **Regresión lineal múltiple:** Utiliza la función fitlm para ajustar el modelo de regresión lineal.
4. **Resultados:** El modelo ajustado se muestra en pantalla, y se calculan los valores estimados de y.
5. **Visualización:** Un gráfico de dispersión compara los valores reales y estimados de y.

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

6. **Estadísticas:** Se muestra el coeficiente de determinación R^2 , que indica la calidad del ajuste.

Ejercicio 101_01. Regresión múltiple (unseen data) con Python

```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Cargar los datos de entrenamiento
data_entrenamiento = pd.read_excel('tu_archivo_entrenamiento.xlsx')
X_entrenamiento = data_entrenamiento.iloc[:, :-1] # Todas las columnas excepto la última
y_entrenamiento = data_entrenamiento.iloc[:, -1] # Última columna

# Crear el modelo de regresión lineal múltiple
model = LinearRegression()
model.fit(X_entrenamiento, y_entrenamiento)

# Cargar los nuevos datos
data_nuevos = pd.read_excel('tu_archivo_nuevos_datos.xlsx')
X_nuevos = data_nuevos

# Realizar las predicciones
predicciones = model.predict(X_nuevos)

# Guardar las predicciones en un nuevo archivo Excel
pd.DataFrame(predicciones).to_excel('predicciones.xlsx', index=False)
```

Explicación:

1. Carga de datos:


- readtable: Esta función lee un archivo Excel y lo carga en una tabla de MATLAB, lo que facilita el manejo de los datos.
- data_entrenamiento: Se crea una variable para almacenar los datos de entrenamiento.
- y_entrenamiento: Se extrae la última columna de la tabla, que corresponde a la variable dependiente (la que queremos predecir).
- X_entrenamiento: Se extraen todas las columnas excepto la última, que corresponden a las variables independientes (las que utilizaremos para hacer la predicción).

2. Creación del modelo:

- fitlm: Esta función crea un modelo de regresión lineal múltiple. Le pasamos como argumentos las variables independientes (X_entrenamiento) y la variable dependiente (y_entrenamiento). Este modelo encontrará la relación lineal entre las variables y creará una ecuación que nos permitirá hacer predicciones.

3. Carga de nuevos datos:

- Se repiten los pasos 1 y 2 para cargar los nuevos datos que queremos predecir.

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

- X_nuevos: Aquí asumimos que todos los datos en el nuevo archivo son predictores, por lo que tomamos todas las columnas.
- 4. **Realización de predicciones:**
 - predict: Esta función utiliza el modelo creado (mdl) para hacer predicciones sobre los nuevos datos (X_nuevos). El resultado se almacena en la variable predicciones.
- 5. **Guardar predicciones (opcional):**
 - writetable: Esta función guarda las predicciones en un nuevo archivo Excel llamado 'predicciones.xlsx'.

Ejercicio 101_02. Regresión múltiple (unseen data) con MATLAB

```
% Cargar los datos de entrenamiento desde el primer archivo Excel
data_entrenamiento = readtable('tu_archivo_entrenamiento.xlsx');
y_entrenamiento = data_entrenamiento{: ,end};
X_entrenamiento = data_entrenamiento{: ,1:end-1};

% Crear el modelo de regresión lineal múltiple
mdl = fitlm(X_entrenamiento,y_entrenamiento);

% Cargar los nuevos datos desde el segundo archivo Excel
data_nuevos = readtable('tu_archivo_nuevos_datos.xlsx');
X_nuevos = data_nuevos{: ,:}; % Asumimos que todos los datos en el nuevo archivo
son predictores

% Realizar las predicciones
predicciones = predict(mdl, X_nuevos);

% Guardar las predicciones en un nuevo archivo Excel (opcional)
writetable(table(predicciones),'predicciones.xlsx');
```

En resumen:

- **Entrenamiento:** Se utiliza un conjunto de datos (entrenamiento) para crear un modelo que relaciona las variables independientes con la variable dependiente.
- **Predicción:** Se utilizan los nuevos datos y el modelo entrenado para estimar los valores de la variable dependiente.

Conceptos clave:

- **Regresión lineal múltiple:** Es una técnica estadística que busca modelar la relación entre una variable dependiente y dos o más variables independientes.
- **Modelo:** Es una representación matemática de la relación entre las variables.
- **Predicción:** Es el proceso de estimar el valor de una variable dependiente para nuevos datos, utilizando el modelo entrenado.