

TP 02: Regression with scikit-learn

Objective

The objective of this practical session is to understand and implement **regression models** using **scikit-learn**. Students will learn how to:

- Prepare and split data
- Train a regression model
- Make predictions
- Evaluate model performance using metrics

Part 1: Theoretical Questions

1. What is **regression** in machine learning?
2. What is the difference between **regression** and **classification**?
3. Write the mathematical model of **linear regression**:

$$y=ax+by = ax + by=ax+b$$

Explain each term.

4. What is the role of the **cost function**?

Part 2: Practical Exercise – House Price Prediction

Problem Description

We aim to predict the **price of a house** based on its **size (m²)** using linear regression.

Step 1: Import Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

Step 2: Create the Dataset

```
# House sizes (m2)
X = np.array([50, 60, 75, 100, 120, 150, 180]).reshape(-1, 1)
```

```
# Prices (in thousands DA)
y = np.array([150, 180, 225, 300, 360, 450, 540])
```

Step 3: Split the Dataset

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

Step 4: Train the Model

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Step 5: Make Predictions

```
y_pred = model.predict(X_test)
```

```
# Predict price for a new house (130 m2)
new_price = model.predict([[130]])
print("Predicted price:", new_price)
```

Step 6: Evaluate the Model

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
print("Mean Squared Error:", mse)
print("R2 Score:", r2)
```

Step 7: Visualization

```
plt.scatter(X, y, color='blue', label='Real Data')
plt.plot(X, model.predict(X), color='red', label='Regression Line')
plt.xlabel("Size (m2)")
plt.ylabel("Price (thousands DA)")
plt.title("House Price Prediction")
plt.legend()
plt.show()
```

