TP3- Testing and Collaboration

Exercise 1: Unit Test for Data Cleaning

Objective: Practice writing unit tests with pytest.
Task:
Write tests for a clean_data(df) function that removes duplicates and nulls.

- Duplicates are removed correctly.
- All null values are dropped.
- The number of rows decreases after cleaning when nulls or duplicates exist.

Exercise 2: TDD - Normalization Function

Objective: Apply Test-Driven Development (TDD).
Task:

1. Write tests *first* for a function normalize_column(df, column) that scales values between 0 and 1.
2. Implement the function to make the tests pass.

   - All normalized values are within [0, 1].
   - Output column length matches input.
   - Invalid column name raises a KeyError.

Exercise 3: Testing Model Evaluation Function

Objective: Test ML evaluation logic using pytest.
Task:
Write tests for evaluate_model(y_true, y_pred) that returns a dictionary with accuracy and F1 score.

- Accuracy = 1.0 for perfect predictions.
- F1 score = 0.0 when all predictions are wrong.
- Output contains both accuracy and f1_score keys.

Exercise 4: Continuous Integration with GitHub Actions

Objective: Automate testing using GitHub workflows.
Task:

1. Create a .github/workflows/run-tests.yml file.
2. Configure it to:

    - Run tests on every push or pull request.
    - Set up Python (version 3.10 or higher).
    - Install dependencies (pytest, pandas, scikit-learn).
    - Execute pytest -v.

Exercise 5: End-to-End Testing (Integration Test)

Objective: Combine testing of multiple components.
Task:
Create and test a mini ML pipeline with three functions:

1. load_data() - loads and returns a DataFrame.
2. train_model() - trains a simple model (e.g., logistic regression).
3. evaluate_model() - returns accuracy.

- Data loads correctly (non-empty, correct columns).
- Model trains without error.
- Accuracy is between 0 and 1.