Module 3: Mini Project-1 Part-B

Bike rental prediction

Testing and Packaging

For this project, we will test and package a bike rental count prediction system using modular programming. Please refer to Module 3 - AST 2 for this mini-project.

PART B [Mini-project Session - 6th Jan 2024, Afternoon]

Step 1: Ensure to go through the previous mini-project [PartA]

Step 2: Project Setup in VS Code: (2 points)

2.1 Use the existing project folder from the previous mini-project PartA session and open it in VS Code.

2.2 Update the project structure by creating new files for testing and packaging, as shown below.

- Add file for input data validation: `processing/validation.py`
- Update the `predict.py` file to include input data validation
- Add the files for testing: `conftest.py`, `test_features.py`, `test_predictions.py`
- Add test requirements
- Add the files related to packaging: `pyproject.toml`, `setup.py`, `manifest.in`, and `mypy.ini`

```
├─Application
│    MANIFEST.in
│    mypy.ini
│    pyproject.toml
│    setup.py
│
│   ├──bikeshare_model
│   │    config.yml
│   │    pipeline.py
│   │    predict.py
│   │    train_pipeline.py
│   │    VERSION
│   │    __init__.py
│   │
│   │   ├──config
│   │        core.py
│   │        __init__.py
│   │
│   │   ├──datasets
│   │        bike-rental-dataset.csv
│   │        __init__.py
│   │
│   │   ├──processing
│   │        data_manager.py
│   │        features.py
│   │        validation.py
│   │        __init__.py
│   │
│   │   └──trained_models
│   │        __init__.py
│   │
│   ├──requirements
│   │    requirements.txt
│   │    test_requirements.txt
│   │
│   └──tests
│        conftest.py
│        test_features.py
│        test_predictions.py
│        __ini__.py
```

Step 3: Implement the following test cases: (3 points)

Implement test cases for:

- Pipeline processing steps, including imputation, mapping, and custom class transformations
- Prediction steps

Step 4: Create a Virtual Environment:

4.1 Open the terminal in VS Code and navigate to the project folder.

4.2 Create a virtual environment as demonstrated in Module 3 - AST 1

Step 5: Install Dependencies: (1 point)

5.1 Activate the virtual environment in the terminal.

5.2 Install the necessary dependencies by running the "pip install" command for required libraries.

Step 6: Train the Model: (1 point)

6.1 Execute the "train_pipeline.py" script to train the bike rental prediction model using the prepared data.

Step 7: Run Test cases: (2 points)

7.1 Run the test cases (created in Step 3) by executing the "pytest" command in the terminal.

Step 8: Build a package for the application: (1 point)

8.1 Install the “build” library by running the "pip install" command.

8.2 Run the "build" command to create distributable files (.tar, .whl, etc).