Module 5: Mini Project-1 Part-C
Bike rental prediction
FastAPI-Deployment

For this project, we will deploy the model built in Project-1 Part-B using FastAPI. Please refer to Module 5 - AST 3 for this mini-project.

PART C [Mini-project Session - 15 March 2025]

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

Step 2: Build a package for the application: (1 point)

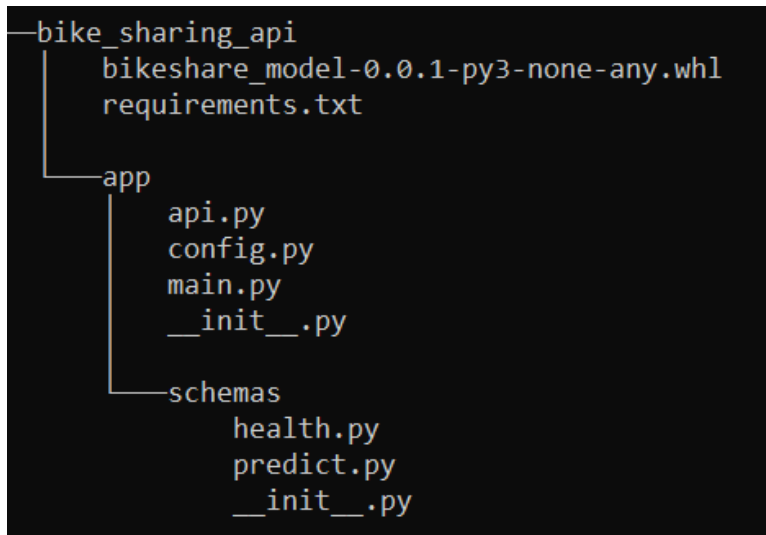2.1 Install the "build" library by running the "pip install" command.

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

2.3 Use the .whl file to import functionalities of the model into the FastAPI project.

Step 3: Project Structure in VS Code: (2 points)

Setup the project structure in VS Code similar to the one shown in below figure.

- Consider 'bike_sharing_api' to be the root directory.
- .whl file from Part-B of the mini project:
  `bike_sharing_api/bikeshare_model***.whl`
- Requirement file: `bike_sharing_api/requirements.txt`
- app folder will contain the files and folders related to FastAPI:
  `bike_sharing_api/app`

```
—bike_sharing_api
    bikeshare_model-0.0.1-py3-none-any.whl
    requirements.txt

    └──app
        api.py
        config.py
        main.py
        __init__.py

        └──schemas
            health.py
            predict.py
            __init__.py
```

Step 4: app folder structure and files: (2 points)

app folder will contain one schemas folder and 4 files:

- __init__.py file: can contain the version number of the app.
- api.py file: This is where API router is defined, make necessary changes to the file.
- config.py file: make necessary changes to the file.
- main.py file: file to be executed to run the app, make necessary changes.

Step 5: Files within schemas folder: (2 points)
- health.py file: name of the app, version of API, and model are validated here.
- predict.py file: contains data schema.
- __init__.py file: define imports from health.py and predict.py

Step 6: Create a Virtual Environment: (1 point)

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

6.2 Create a virtual environment as demonstrated in Module 5 - AST 1

Step 7: Install Dependencies: (1 point)

7.1 Activate the virtual environment in the terminal.

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

Step 8: Start the Application: (1 point)

8.1 Execute the "main.py" script to start the application and get the server url.

8.2 Make one prediction and test.