

## **Module 3: Mini Project-1 Part-A**

### **Bike rental prediction**

### **Regression and Modularization**

For this project, we will build a bike rental count prediction system using modular programming. Please refer to Module 3 - AST 1 for this mini-project.

#### **PART A [Mini-project Session - 6th Jan 2024, Morning]**

##### **Step 1: Understanding the ML Workflow: (4 points)**

1.1 Study Notebook-1 (code given):

- Understand the different steps involved in the notebook, including Dataset description, Data loading, EDA, Data Preprocessing, Feature Engineering, Model Building, Training, and Prediction.

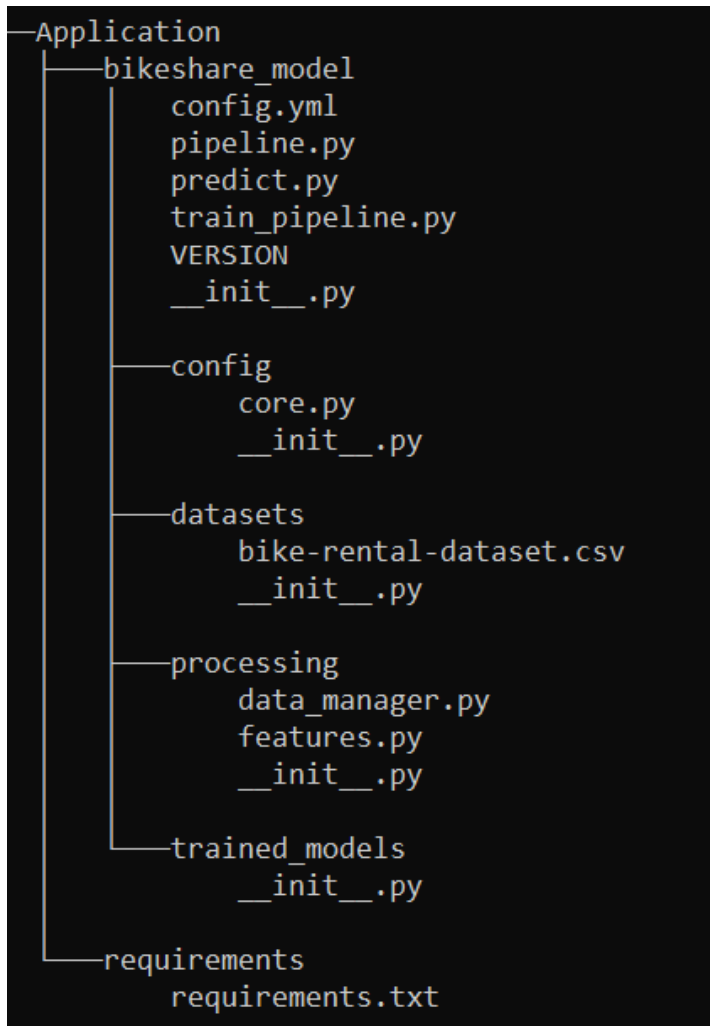
1.2 Study Notebook-2 (no code provided): Develop the code for following steps:

- Data loading and Pre-pipeline processing
- Pipeline processing including imputation, mapping, and custom class transformations
- Final pipeline building and training

##### **Step 2: Project Setup in VS Code: (1 point)**

2.1 Create a new project folder and open it in VS Code.

2.2 Create the project structure and organization similar to below.



### Step 3: Implement the Pipeline Building Steps: (2 points)

Use code implementation from Notebook-2 to implement:

- Data loading and pre-pipeline processing steps in Python modular files.
- Pipeline processing steps, including imputation, mapping, and custom class transformations, in Python modular files.
- Pipeline training steps
- Prediction steps

### Step 4: Create a Virtual Environment: (1 point)

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 and Generate Predictions: (1 point)**

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

6.2 Evaluate the model's performance and tune parameters to obtain a better accuracy as required.

6.3 Run the "predict.py" script to generate predictions based on new input data.