**MLOps Exercise: Batch Processing of Bike Sharing Data**

**Objective**

In this exercise, you will build a batch processing pipeline that uses a pre-trained machine learning model to make predictions on bike-sharing batch data. You will then save the results of the predictions. This exercise will help you:

1. Understand how to work with batch data.
2. Automate predictions using a machine learning model.

**Dataset**

You will use the bike-sharing batch data. Each record represents various features related to bike usage, such as:

* **hour**: Hour of the day the record is for.
* **season**: Season of the year (e.g., 1 = winter, 2 = spring, etc.).
* **holiday**: Whether it's a holiday (1 = yes, 0 = no).
* **temp**: Normalized temperature.
* **[Your task is to predict the column]:** **count** (number of users for a given hour).

**Write a Batch Processing Script:**

1. **Create a Python batch processing that:**
   * **Reads the batch data from a given input CSV file.**
   * **Loads the pre-trained bike-sharing model (provided as a pickle file).**
   * **Runs predictions on the batch data and adds the predictions as a new column.**
   * **Saves the updated batch data (with predictions) to a new CSV file (output/predictions.csv).**

**Notes:**

* **Ensure the script accepts the input file and output file as arguments when the script is executed.**
* **Try to write the script as OOP as your can (split to different classes and funcs)**
* **Make the script Flexible to changes (new model, new data source)**

**Bonus #1: Dockerize the Solution**

Containerize the entire batch processing pipeline using Docker. You'll create a Dockerfile that:

* Sets up a Python environment with all required dependencies.
* Copies the batch processing script and necessary assets (model file, input data) into the container.
* Runs the batch processing script automatically.

Expected Workflow inside the Docker container:

* The input batch file (input/bike\_sharing\_batch.csv) is provided.
* Predictions are saved to output/predictions.csv.

**Bonus #2: PostgreSQL Integration**

Use a PostgreSQL database as input instead of a CSV file:

1. Write code to load the batch data from a PostgreSQL table instead of reading from a CSV file.
2. Store the predictions back into a new table in the PostgreSQL database.

Steps:

1. Set up a PostgreSQL database as a Docker container.
2. Create a table **bike\_sharing\_data** with relevant columns
3. Populate the table with example data.
4. Update your Python batch processing script to query data from this table, run predictions, and then save the output back into another table (e.g., **bike\_sharing\_predictions**).