Code Production System

Project Structure

```
Question 3/
├── src/
│   ├── data_handler.py
│   ├── data_preprocessor.py
│   ├── model_evaluator.py
│   ├── model_trainer.py
│   └── predictor.py
├── data/
│   ├── historical_sensor_data.csv
│   └── latest_sensor_data.csv
├── main.ipynb
└── requirements.txt
```

Description

This project is designed to handle, preprocess, train, evaluate, and predict using sensor data. The main components of the project are as follows:

src/data_handler.py: Contains functions to load and manage data.
src/data_preprocessor.py: Contains functions to preprocess the data.
src/model_trainer.py: Contains functions to train the model.
src/model_evaluator.py: Contains functions to evaluate the model.
src/predictor.py: Contains functions to make predictions using the trained model.
data/historical_sensor_data.csv: Historical sensor data used for training and evaluation.
data/latest_sensor_data.csv: Latest sensor data for making predictions.
main.ipynb: Jupyter Notebook to demonstrate the usage of the modules.
requirements.txt: Contains the list of dependencies required for the project.

Setup

1**. Clone the repository**

git clone <repository_url>

cd Question 3

2. Create and activate a virtual environment

python -m venv venv

source venv/bin/activate # On Windows, use `venv\Scripts\activate`

3. Install the dependencies

pip install -r requirements.txt

Usage

1. Load and preprocess the data: The data handler and preprocessor scripts are used to load and preprocess the historical sensor data.
2. Train the model: Use `model_trainer.py` to train the model on the preprocessed data.
3. Evaluate the model: Use `model_evaluator.py` to evaluate the performance of the trained model.
4. Make predictions: Use `predictor.py` to make predictions on the latest sensor data.

All these steps are demonstrated in the `main.ipynb` notebook.