1. Machine Learning

Input Data: bikedata.csv

Code:Bike\_data.ipynb: Install Holidays to check if that day is holiday

2. Deep Learning

Input Data: Bike infor.csv

Code: DLmodels.ipynb: Install PyTorch to predict the in/out bike

3. Optimization Model

Input Data: Ensure you have the necessary input data files: station\_ID\_Capacity.xlsx, distance\_matrix.xlsx, LSTMresult.csv, and optimization\_result.xlsx.

Output: The script will display the optimal bike transfer solutions, indicating the number of bikes to be transferred between stations to minimize total distance.

* Model.py: The main Python script.
* station\_ID\_Capacity.xlsx: Excel file containing station IDs and capacities.
* distance\_matrix.xlsx: Excel file with the distance matrix between stations.
* LSTMresult.csv: CSV file with LSTM model predictions for bike inflow and outflow.
* ground\_truth.csv: CSV file containing actual numerical values representing bike inflow and outflow for each station.
* optimization\_result.xlsx: Excel file is generated by calculating the results from the Optimization Model and comparing them with the ground truth. It contains several sheets:
* 7.2(O) … 7.9(O): Excel sheets with the number of bikes in each station after moving and adjusting based on the original number of bikes. This is also part of the input for the next day’s optimization.
* 7.2(C) … 7.7(C): Excel sheets with the number of bikes in each station after moving and adjusting based on the capacity. This is also part of the input for the next day’s optimization.