group04

Start the Jupyter Notebook through Docker (see instructions at below section).

Generate the entire analysis report through `make all` command

In the Terminal window in Jupyter Notebook, run the following command:

```
make all
```

The generated images and models that are used in the report should now be available in the following folders:

```
./results/images/
./results/models/
```

The generated HTML file should now be available in the following folder:

```
./reports/ttc_bus_delay_report.html
```

Undo the above process through `make clean` command

To remove the images and models generated by the above process, run the following command:

```
make clean
```

Start the Jupyter Notebook through Docker (see instructions at below section).

Running preprocesing script

In the Terminal window in Jupyter Notebook, run the following commands:

Navigate to the scripts folder

```
cd scripts
```

Run the preprocess.py script by using the following command in the terminal:

```
python preprocess.py --raw_data ~/data/ttc-bus-delay-data-2024.csv --preprocessed_data ~/data
```

There are multiple command line arguments required to run the script successfully, and those must be provided as is to run the script, or to create folders

Running data validation script

```
python ttc_data_validation.py --input-path ../data/clean/X_train.csv --output-path ../data/cl
```

Running eda script

```
python ttc_eda.py --input-path ../data/clean/ttc-bus-delay-clean.csv --output-dir ../results/
```

Running analysis script

The analysis file also has multiple command line arguments which must me run from the scripts folder. The command to run the analysis.py file is:

```
python analysis.py --data ~/data/clean --preprocessor_from ~/results/models/delay_preprocesso
```

Generating analysis report through quarto document

Start a new Terminal window in the Jupyter Notebook. Navigate to the reports folder

```
cd reports
```

Run the following command to generate the HTML report file.

```
quarto render ttc_bus_delay_report.qmd --to html
```

The generated HTML file should now be available in the following folder:

```
./reports/ttc_bus_delay_report.html
```

Docker Container Setup

Prerequisites

- Docker
- Docker Compose

Use the Docker Image

Pull the Docker Image from DockerHub

To pull the latest version of the Docker image from DockerHub, use:

```
docker pull agam007/group04:latest
```

Run the Docker Container to start Jupyter Notebook

Next, start a container and map port 8888 for Jupyter Notebook access. The command is:

```
docker run \
    -it \
    --rm \
    -p 8888:8888 \
    -v .:/home/jovyan \
    agam007/group04:latest \
    start-notebook.sh \
    --NotebookApp.token='' \
    --NotebookApp.password=''
```

Go to http://localhost:8888/ to access the Jupyter Notebook.

Update the Docker Image on DockerHub

If any changes are made to the environment files or Docker configuration files in this repository, the image on DockerHub will be automatically updated through the Github Actions Workflow.

Use Docker Compose (Recommended)

Another simpler way to launch and manage containers is to use Docker Compose.

To start the services defined in the `docker-compose.yml` file, use:

```
docker-compose up
```

Similar to the above, go to http://localhost:8888/ to access the Jupyter Notebook.

To stop the services, press `Ctrl+C` in the terminal where `docker-compose up` is running, or use:

```
docker-compose down
```

Project Title

Toronto TTC Bus Delay Report

Description

This project aims to analyze the delay time (in minutes) for various bus routes in Toronto and build a model to predict future delays based on explanatory variables, including: - Day of the week - Month - Type of incident (if any) - Minimum delay time recorded

We build a predictive model using historical bus data in 2024 to determine the likelihood and extent of delays for future bus operations.

Table of Contents