

Project Documentation

Deployment Process

Prerequisites

1. **System Requirements:** Ensure you have a system with the following:
 - Python 3.8 or above
 - Docker installed (optional for containerized deployment)
 - pip (Python package manager)
2. **Libraries/Dependencies:**
 - Install the required Python packages listed in the `requirements.txt` file using:

```
pip install -r requirements.txt
```
3. **Model Files:**
 - Ensure the following model files are available in the project directory:
 - `linear_regression_model.pkl`
 - `rfe_data_model.pkl`
4. **Datasets:**
 - Include the following datasets in the project directory:
 - `lasso_selected_data.csv`
 - `rfe_selected_features.csv`
 - `test.csv`

Deployment Steps

Option 1: Local Deployment

1. **Clone the Repository:**

```
git clone <repository_url>  
cd <repository_name>
```

2. **Run the Application:**

```
python app.py
```

This will start the server on `http://127.0.0.1:5000` by default.

Option 2: Dockerized Deployment

1. Build the Docker Image:

```
docker build -t flask-regression-app.
```

2. Run the Docker Container:

```
docker run -p 5000:5000 flask-regression-app
```

Access the application at `http://localhost:5000`.

Usage Instructions

Interacting with the Web Application

- Access the Web App:** Open a browser and navigate to `http://127.0.0.1:5000` or the appropriate Docker container address.
- Upload Dataset:**
 - Use the upload feature to input datasets (e.g., `test.csv`).
- Run Predictions:**
 - Click the "Predict" button to analyze the dataset.
- View Results:**
 - View predictions and insights directly on the dashboard.

Input:

Delivery Time Prediction

Upload CSV

test.csv

Manual Input

Actual Distance to Destination

Segment OSRM Time

OD Duration (Hour)

OSRM Distance Time Ratio

Output:

The output will have the entered data with the data that has been predicted which here is the delivery time

data	route type	start scan to end scan	actual distance to destination	segment actual time	segment osrm time	od duration dirr hour	osrm distance time ratio	distance time ratio	segment actual time sum	predicted delivery time
2024-12-27	Urban	10:00-10:15	15.000000	12.5	13.2	0.800000	1.050000	0.900000	14.0	13.765838
2024-12-27	Rural	10:15-10:30	25.500000	20.0	22.0	1.000000	1.100000	1.000000	23.0	24.134693
2024-12-27	Suburban	10:30-10:45	12.000000	10.5	11.0	0.600000	0.980000	0.950000	11.5	10.702691
2024-12-27	Urban	10:45-11:00	8.500000	6.8	7.0	0.500000	1.000000	1.100000	7.5	7.064030
2024-12-27	Rural	11:00-11:15	18.000000	15.5	16.2	0.900000	1.080000	0.980000	17.0	16.691859
1.0	0.0	86.0	10.435660	14.0	11.0	1.436894	1.087755	-3.564340	14.0	13.732366
1.0	0.0	86.0	18.936842	10.0	9.0	1.436894	1.086215	-5.063158	24.0	23.548231
1.0	0.0	86.0	27.637279	16.0	7.0	1.436894	1.162125	-12.362721	40.0	39.391393
1.0	0.0	86.0	36.118028	21.0	12.0	1.436894	1.139050	-25.881972	61.0	61.292089
1.0	0.0	86.0	39.386040	6.0	5.0	1.436894	1.232230	-28.613960	67.0	67.169446
1.0	0.0	109.0	10.403038	15.0	11.0	1.819553	1.101555	-4.596962	15.0	14.741021
1.0	0.0	109.0	18.045481	28.0	6.0	1.819553	1.252294	-25.954519	43.0	43.580471
1.0	0.0	109.0	28.061896	21.0	11.0	1.819553	1.235352	-36.938104	64.0	64.458772

API Details

Base URL

- Local: <http://127.0.0.1:5000>
- Docker: <http://localhost:5000>

Endpoints

1. /

- Method:** GET
- Description:** Renders the homepage of the web application.

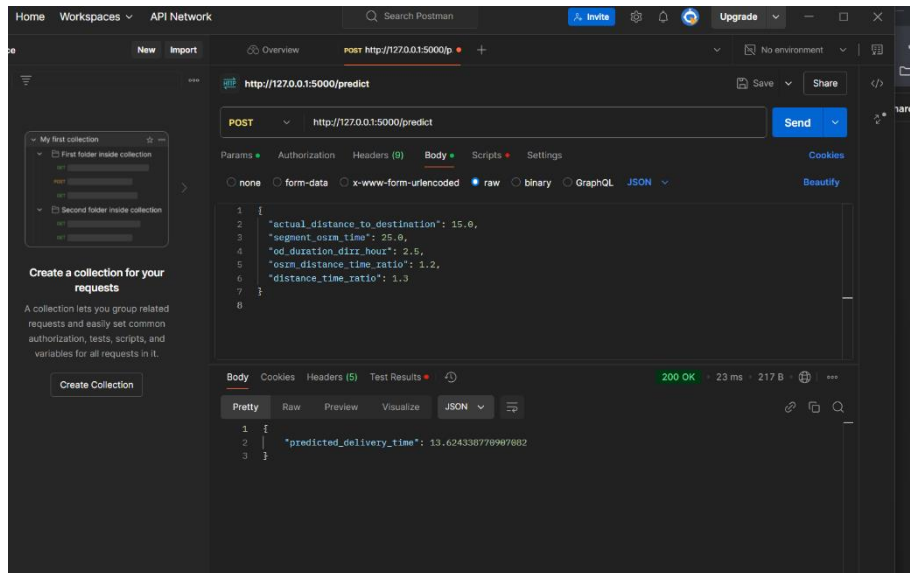
2. /predict

- Method:** POST
- Description:** Processes the uploaded dataset and returns predictions.
- Payload:**

```
{  
  Json format data  
}
```

- **Response:**

```
{  
  "predictions": [<list_of_predictions>]  
}
```



User Guide

Steps for Interacting with the Web Application

1. Uploading a File:

- Navigate to the "Upload" section.
- Browse and select a CSV file (e.g., test.csv).
- A person can input the preferred data manually.

2. Submit for Prediction:

- Click on the "Predict" button to initiate the analysis.

3. View and Download Results:

- Results are displayed in a table format on the dashboard.
- Use the "Download" button to save the results.

Troubleshooting

1. Common Issues:

- **Error:** Missing file or dataset.
 - **Solution:** Ensure required files (e.g., `linear_regression_model.pkl`) are in the correct directory.
- **Error:** Dependency issues.
 - **Solution:** Reinstall dependencies using `pip install -r requirements.txt`.