

HARDIKKUMAR SUTARIYA

Wireframe Document

Metro Interstate Traffic Prediction



Contents

| | |
|--|----|
| Document Version Control | 2 |
| 1. Introduction | 4 |
| 1.1 Why this Wireframe Document? | 4 |
| 1.2 What is the scope? | 4 |
| 2. Prediction Webpage..... | 5 |
| 2.1 Homepage..... | 5 |
| 2.2 Giving form inputs..... | 6 |
| 2.3 Prediction... .. | 6 |
| 2.4 Testing with incorrect ranged input | 7 |
| 3. API Response..... | 8 |
| 3.1 Homepage after deploying the app | 8 |
| 3.2 Testing with Postman | 9 |
| 3.3 Providing input in json format..... | 10 |
| 3.4 Prediction using API | 11 |
| 3.4 Testing API Response with intended exception | 12 |

1 Introduction

What is Wireframe Document?

A wireframe is a document that outlines the structure of a website or app. It's a tool that clearly indicates what needs to be on every page. More importantly it establishes the relationship between everything on each page.

What is Scope?

Wireframing is a way to design a website service at the structural level. A wireframe is commonly used to layout content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content is added.

2 Prediction Webpage

2.1 Homepage

Homepage of the deployed app.

Metro Interstate Traffic Prediction Home source code

Enter the details as indicated:

Holiday expected range 0 and 1

Temperature expected range 243.39 to 310.07

Cloud Percentage expected range 0 to 100

Weather Info

Month expected range 1 to 12

Weekday expected range 0 to 6

Current Hour expected range 0 to 23

Predict

Prediction:

Weather Category:

Clear : 0
 Clouds : 1
 Drizzle : 2
 Fog : 3
 Haze : 4
 Mist : 5
 Rain : 6
 Smoke : 7
 Snow : 8
 Squall : 9
 Thunderstorm : 10

- **Home** – Homepage for Metro Interstate Traffic Prediction app.
- **Source Code** – Redirects to the GitHub source code page.
- **Input fields** – Input with respect to model trained features .
- **Input ranges** – Placeholder guiding the valid ranges for input. Any values outside these ranges will lead to the error page.
- **Prediction** – Prediction will be displayed here.
- **Weather Category** – Defines the assigned numbers for weather types as per label encoded features.
- **Predict** – Button to display result of prediction.

2.2 Giving form inputs

Metro Interstate Traffic Prediction [Home](#) [source code](#)

Enter the details as indicated:

Predict

Prediction:

Weather Category:

Clear : 0
Clouds : 1
Drizzle : 2
Fog : 3
Haze : 4
Mist : 5
Rain : 6
Smoke : 7
Snow : 8
Squall : 9
Thunderstorm : 10

The values entered in each field should lie between the valid input range, otherwise upon prediction the application will be redirecting the error page with the respective error.

2.3 Prediction

The Predicted result are too ranged between appropriate analyzed values, but as far as the input value is within the domain range, the prediction out of range exception is less likely to be triggered.

Metro Interstate Traffic Prediction [Home](#) [source code](#)

Enter the details as indicated:

Predict

Prediction:

Weather Category:

Clear : 0
Clouds : 1
Drizzle : 2
Fog : 3
Haze : 4
Mist : 5
Rain : 6
Smoke : 7
Snow : 8
Squall : 9
Thunderstorm : 10

2.4 Testing with incorrect ranged input

The application is tested with intended exceptions such as invalid columns, invalid datatypes and invalid ranges.

[Metro Interstate Traffic Prediction](#) [Home](#) [source code](#)

Enter the details as indicated:

0

100

0

12

13

6

29

Predict

Prediction:

Weather Category:

Clear : 0
Clouds : 1
Drizzle : 2
Fog : 3
Haze : 4
Mist : 5
Rain : 6
Smoke : 7
Snow : 8
Squall : 9
Thunderstorm : 10

Oops!

404 Not Found

ERROR: Values entered are not in range!

Take Me Home

3 API Response

3.1 Homepage after deploying app

Predicting traffic volume through API response is the second feature of our prediction app. The deployed app link can be used for API response.

The screenshot shows a web browser window displaying the 'Metro Interstate Traffic Prediction' app. The browser's address bar shows the URL 'https://traffic-prediction-app.onrender.com'. The app's header includes the title 'Metro Interstate Traffic Prediction' and links for 'Home' and 'source code'. The main content area is divided into three sections: 'Enter the details as indicated:', 'Prediction:', and 'Weather Category:'. The 'Enter the details as indicated:' section contains seven input fields with placeholder text: 'Holiday expected range 0 and 1', 'Temperature expected range 243.39 to 310.07', 'Cloud Percentage expected range 0 to 100', 'Weather Info', 'Month expected range 1 to 12', 'Weekday expected range 0 to 6', and 'Current Hour expected range 0 to 23'. Below these fields is a blue 'Predict' button. The 'Prediction:' section is currently empty. The 'Weather Category:' section displays a list of weather categories and their corresponding values: Clear : 0, Clouds : 1, Drizzle : 2, Fog : 3, Haze : 4, Mist : 5, Rain : 6, Smoke : 7, Snow : 8, Squall : 9, and Thunderstorm : 10. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons, including File Explorer, Edge, and Word. The system tray on the right indicates a temperature of 30°C, 'Mostly sunny' weather, and the time 15:13 on 01-01-2023.

Metro Interstate Traffic Prediction Home source code

Enter the details as indicated:

Holiday expected range 0 and 1

Temperature expected range 243.39 to 310.07

Cloud Percentage expected range 0 to 100

Weather Info

Month expected range 1 to 12

Weekday expected range 0 to 6

Current Hour expected range 0 to 23

Predict

Prediction:

Weather Category:

Clear : 0
Clouds : 1
Drizzle : 2
Fog : 3
Haze : 4
Mist : 5
Rain : 6
Smoke : 7
Snow : 8
Squall : 9
Thunderstorm : 10

Type here to search

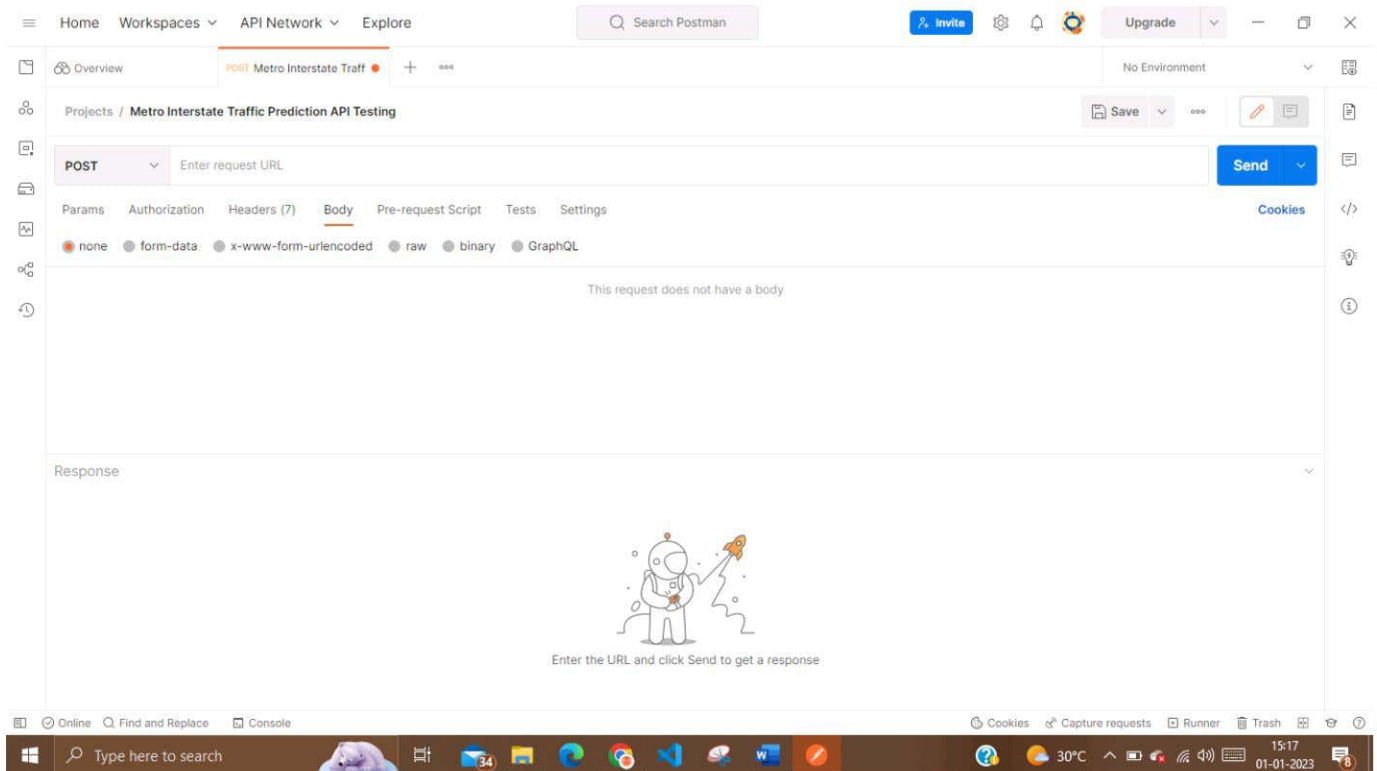
34

30°C Mostly sunny

15:13
01-01-2023

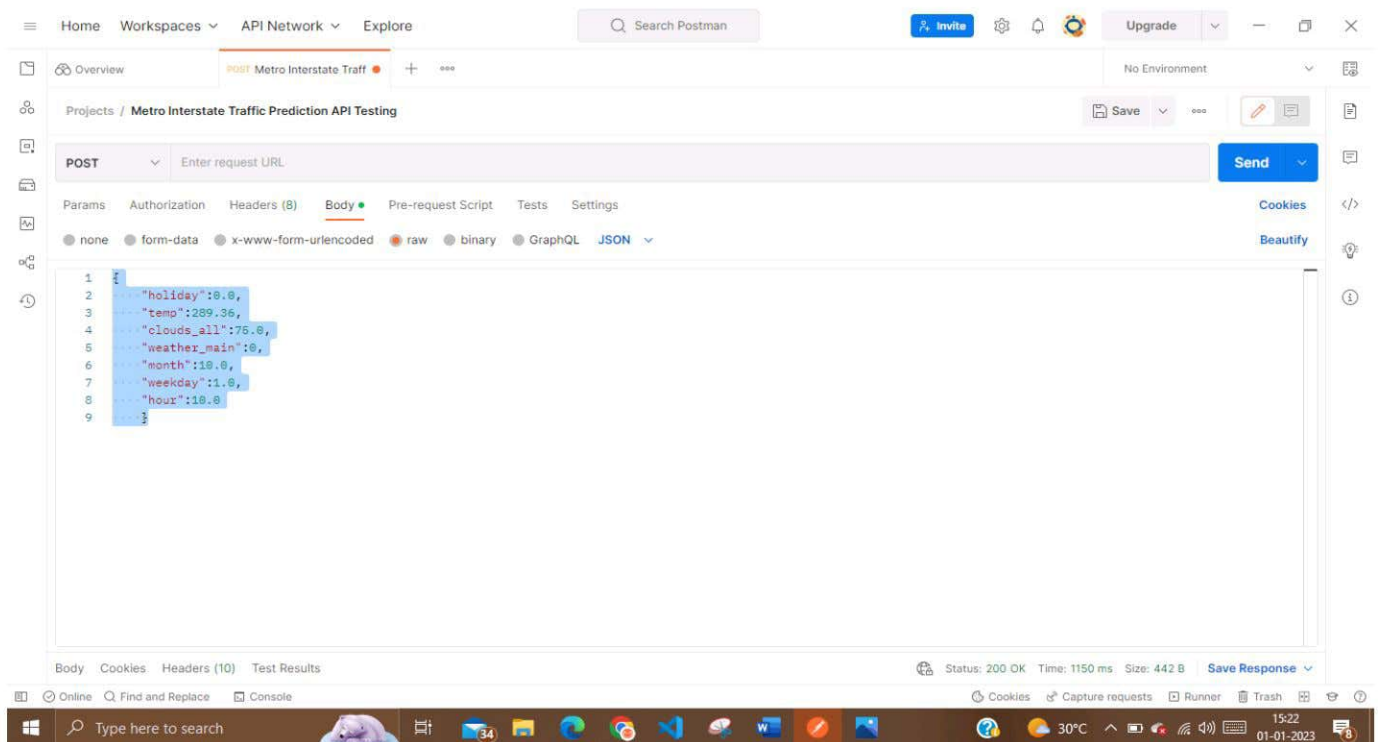
3.2 Testing with Postman

For testing Prediction using API request we will use Postman. Postman is an API platform for developers to design, build, test and iterate their APIs.



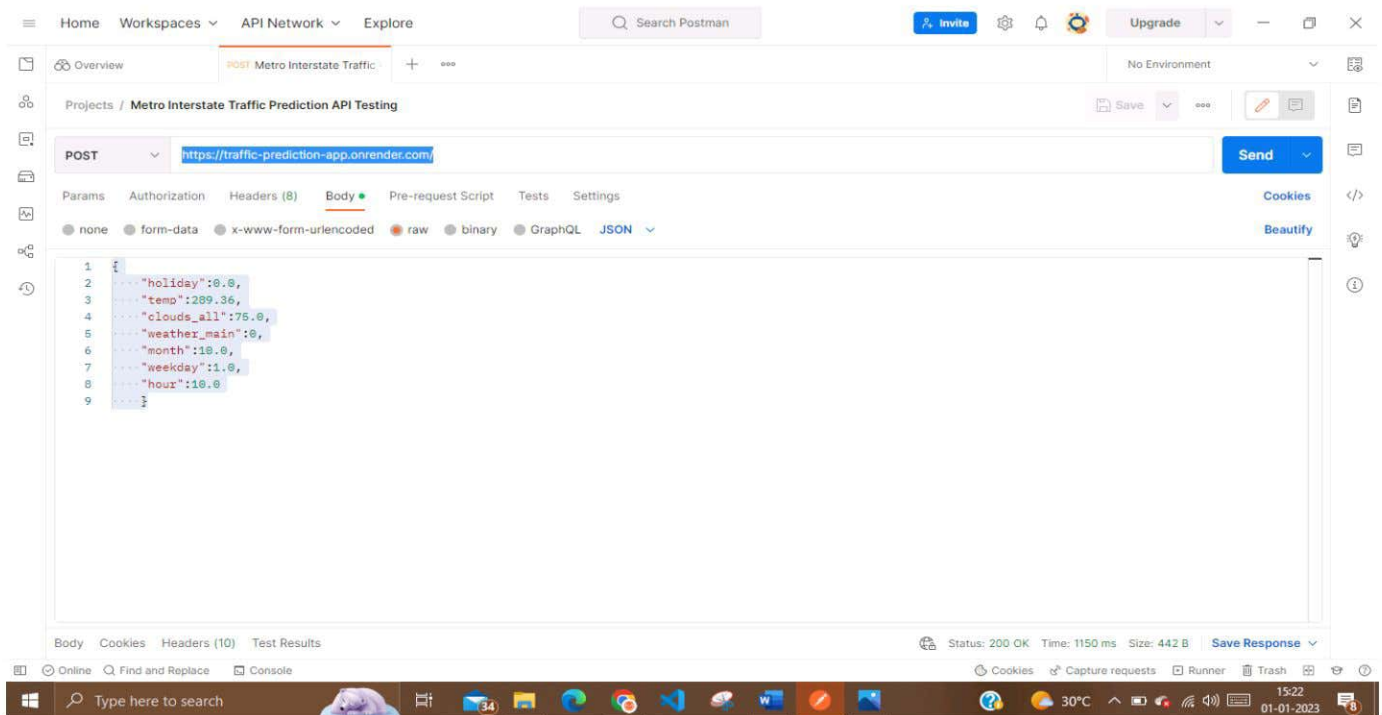
3.3 Providing input in json format

As per our model, for API response prediction we accept only json formatted response. So go to Body>raw and write json formatted inputs.

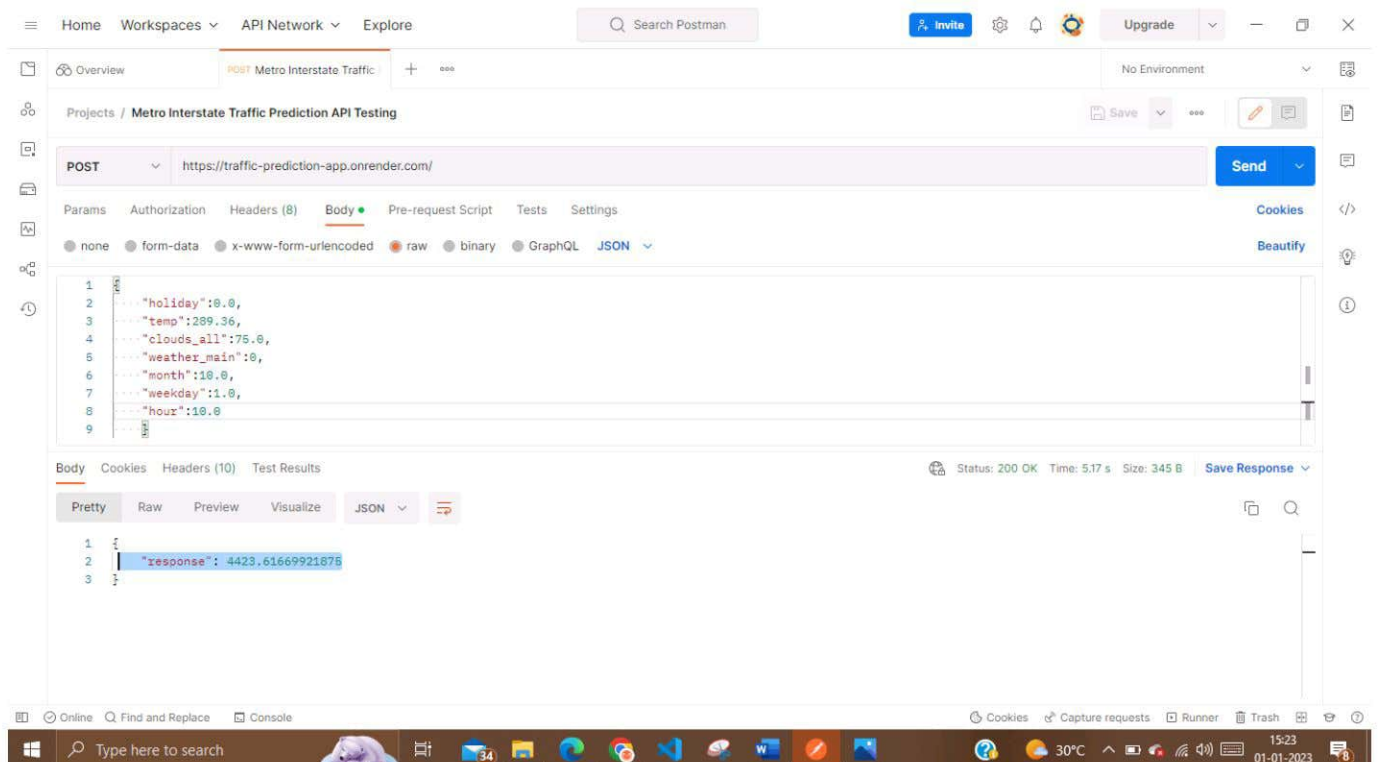


3.4 Prediction using API

After providing the correctly ranged inputs, we will use the deployed app link as an API for returning prediction response from the Postman.



Click on send to get the prediction result.



3.5 Testing API response with an intended exception

Here we will test the prediction response with an intended irrelative column in our json input.

The image displays two screenshots of the Postman application interface, illustrating the setup and execution of an API test.

Top Screenshot: The interface shows a POST request to `https://traffic-prediction-app.onrender.com/`. The request body is a JSON object with the following structure:

```
1 {
2   "holiday": 0.0,
3   "temp": 289.36,
4   "Invalid_column": 75.0,
5   "weather_main": 0,
6   "month": 10.0,
7   "weekday": 1.0,
8   "hour": 10.0
9 }
```

The status bar at the bottom indicates a successful response: `Status: 200 OK`, `Time: 5.17 s`, and `Size: 345 B`.

Bottom Screenshot: This screenshot shows the same POST request, but with the response body displayed in the "Test Results" tab. The response is a JSON object indicating an invalid column:

```
1 {
2   "response": "Invalid columns!",
3   "the_expected_columns": [
4     "holiday",
5     "temp",
6     "clouds_all",
7     "weather_main",
8     "month",
9     "weekday",
10    "hour"
11  ]
12 }
```

The status bar at the bottom indicates a successful response: `Status: 200 OK`, `Time: 1339 ms`, and `Size: 442 B`.