

Key Insights - Daily Public Transport

- The rapid route is being used the most on weekdays, especially Wednesday and Thursday, while rarely being used on weekends, even rarely on Sundays especially.
- From the given heatmap in the attached notebook, we can infer that the rapid route and light rail have a very strong positive correlation (0.97), catering to similar commuter demands, indicating that a change in one of these can largely affect the other.
- While peak service is highly preferred on weekdays, its demand drops to zero on weekends, this might suggest that the peak service is only being operated on weekdays.
- Among all the other routes, the local route has the most stable usage, indicating stable demand among commuters. However it sees some occasional dips (<25), which could be due to other factors like holidays, etc. Despite that, the local route is the route with the most demand, catering to everyday needs of the commuter.

Note: All plots for the data are given in the lpython notebook in the repo.

Forecasting model used - Prophet

Why Prophet?

Prophet is a robust and easy-to-use forecasting tool developed by Facebook specifically for time series data that has:

- **Clear seasonality** (daily/weekly/monthly)
- **Missing data**
- **Irregular intervals**
- **Strong trend components**

It's particularly well-suited for **business time series** like ridership, sales, or service demand — which often follow **weekly patterns** and are affected by **holidays or special events**.

In this project, Prophet was chosen because:

- The data has **weekly patterns** (e.g., ridership drops on weekends)
- The demand trend is **non-linear**
- Forecasting was needed for **multiple categories** (Local Route, Light Rail,

- etc.)
 - The goal was to **predict for the next 7 days**, which Prophet handles very efficiently
-

Model Overview

Prophet decomposes time series data into 3 main components:

$$y(t) = \text{trend}(t) + \text{seasonality}(t) + \text{holidays}(t) + \text{error}$$

1. **Trend**: Captures long-term increase or decrease.
2. **Seasonality**: Handles recurring patterns (like weekly usage).
3. **Holidays**: Optional, but allows inserting known events.
4. **Error**: Captures randomness.

Prophet automatically detects **weekly seasonality**, and we can enable **daily or yearly** seasonality too.

Model Parameters Used

- Default seasonality mode: **additive**
 - Automatically detected **weekly seasonality**
 - Did **not** use holiday effects (not applicable here)
 - Did **not** manually tune changepoints (default behavior works well for short-term forecasting)
-

Output Explained

Prophet returns the following:

- **ds**: Future date
- **yhat**: Predicted value (e.g., number of passengers)
- **yhat_lower**, **yhat_upper**: Confidence interval bounds

Summary

Prophet was chosen for its:

- Ease of use
- Automatic handling of trends & seasonality
- Clean forecast output
- Strong performance on short-term forecasting

This makes it ideal for forecasting transport route usage patterns with minimal manual tuning.

Note: **The plots and predicted values are in the attached notebook**