Technical Report: Linear Regression for Ridership Forecasting

1 Introduction

This report outlines the linear regression algorithm used to forecast daily ridership for the ‘Local Route‘ transportation mode, based on a dataset spanning 2019 to 2024. The model predicts ridership using temporal and aggregate features, achieving high accuracy.

2 Algorithm Description

Linear regression models the relationship between a dependent variable y (ridership) and independent variables X (features) as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

where β_0 is the intercept, β_i are coefficients, and ϵ is the error term. The model minimizes the mean squared error (MSE) using ordinary least squares.

3 Data and Features

The dataset contains 1,918 daily records with columns: ‘Date‘, ‘Local Route‘, ‘Light Rail‘, ‘Peak Service‘, ‘Rapid Route‘, ‘School‘, and ‘Other‘. Features used:

- **month**: Month of the year (1–12).
- **weekday_num**: Day of the week (1=Sunday, 7=Saturday).
- **day**: Day of the month (1–31).
- **Total**: Sum of ridership across all modes.

The ‘year‘ was scaled using ‘MinMaxScaler‘ but not used in the final model.

4 Model Parameters

The ‘LinearRegression‘ model from ‘scikit-learn‘ was configured with:

- **fit_intercept=True**: Includes an intercept term.
- **normalize=False**: No internal normalization (external scaling applied to ‘year‘).
- **n_jobs=None**: Uses single-threaded computation.

No hyperparameter tuning was performed, as linear regression has minimal tunable parameters.

5 Performance

The model was trained on a 75:25 train-test split, achieving:

- **MSE**: 635,466.92, indicating moderate prediction errors.
- **R^2**: 0.983, suggesting 98.3% of variance in ‘Local Route‘ ridership is explained.

A sample prediction (‘month=7‘, ‘weekday$_{n}um = 2‘, ‘day = 1‘, ‘Total = 51,294.1‘)yielded 16,351.89 riders.$

6 Conclusion

Linear regression effectively models ‘Local Route‘ ridership but requires extension to other modes and a 7-day forecast. Future improvements could include feature engineering (e.g., holidays) and alternative algorithms (e.g., Random Forest).