Subway Science: Forecasting New York City's hourly subway ridership with machine learning

Jack Carlisle and Nick Haubrich
Github: https://github.com/nhaubrich/SubwayScience

Overview

With over 400 stations and up to millions of daily riders, the New York City subway system is one of the largest and busiest services of its kind. Accurate forecasts of the subway's usage can benefit individuals for planning trips, rideshare companies for adapting prices, and the MTA itself for allocating resources to the system. In this project, we implement machine learning techniques to model the hourly ridership of the NYC subway.

Dataset and Features

Our dataset consists of hourly ridership at each of the 428 MTA stations from February 2022 to October 2024, and was obtained from data.ny.gov through the MTA Open Data Program. The preprocessing step accumulates and orders the data into a single row per hour with columns for each station.

Altogether, we have 23349 hours of observations, with ridership for each of the 428 stations. Models were trained and validated on the first 70% and 10% of the dataset, then evaluated on the final 20%. Variables corresponding to the time of day, day of the week, and month of the year were included as features via Fourier encoding. A binary flag for national holidays was computed by the holidays package.

Methodology

All models were trained to predict the next hour of subway ridership given the features of the previous 12 hours. Key performance metrics were chosen to be the root mean squared error (RMSE) for predictions 1 hour in advance and 12 hours in advance. Predictions beyond the initial hour were obtained by using predictions as input data in an autoregressive manner.

As an initial task, we developed models to predict the total subway ridership. Results were compared between a linear model (Linear), a dense neural network (NN), and a Long Short-Term Memory Network (LSTM). The LSTM achieved the best performance, as measured by RMSE.

As a second task, models were trained to predict the ridership at each of the 428 stations. The three previous models were adapted to take as input the ridership and time features of the 428 stations over the previous 12 hours, and produce as output a prediction of the next hour's ridership for each of the 428 stations. However, this led to poor performance relative to the total ridership models due to overfitting on the multiple time series. To counteract this, we developed a new neural network that takes as input the ridership of each station individually, plus the time features and a 16-dimensional encoding of a station identifier, and returns as output a prediction of just that station's ridership. Implementing the network as a 1D CNN, with features corresponding to channels and stations corresponding to the stride dimension, allowed for efficient training and yielded greatly improved results.

	Total Ridership			Per-Station Ridership			
Model	Linear	NN	LSTM	Linear	NN	LSTM	CNN
RMSE at 1 hour	0.29	.07	.06	28	28	32	5.2
RMSE at 12 hours	.52	.20	.19	18	19	28	10

Conclusions and Future Directions

We have implemented both a LSTM network which accurately predicts total MTA ridership, and a 1-dimensional CNN which accurately predicts per-station MTA ridership. The total ridership LSTM has an average error of 6400 people (5%) on predictions one hour in advance, and an average error of 21000 people (16%) 24 hours in advance. The per-station CNN model achieves a total RMSE of 5.2 at 1 hour, and a total RMSE of 10 at 12 hours. This corresponds to an average RMSE of .01 and .02 for each station.

The CNN's performance was the worst for the stations servicing Citi Field, Yankee Stadium, and Rockaway beach. These three locations are subject to irregular surges in ridership (due to MLB games or nice weather). Further feature engineering may improve the model. For example, incorporating the home schedule of games and weather predictions (e.g. temperature and precipitation) may substantially improve the model's accuracy for these stations.

Finally, there may be room for further architecture improvements. As the LSTM performed the best for the total ridership case, it is expected that training a LSTM on each station individually would perform as well or better than the CNN. However, this is computationally taxing. It may be possible to incorporate RNN elements into the CNN to obtain a favorable tradeoff between accuracy and computation.