Exploring and Modelling Passenger Flow and crowding in the London Underground Network (pre- and post-COVID)

CID: 01382316

Github Repo: https://github.com/Shawn0511/TFL-underground-network/releases/tag/v.1.0.1

1 Project Description

London's multi-modal rail system moves more than four million passengers on a typical weekday, yet crowding remains unevenly distributed across space and time. This project analyses nine years (2016–2023) of Transport for London (TfL) data to explore how passenger flows and crowding patterns have changed across the London Underground (pre- and post-COVID). The study leverages several public datasets provided by TfL, including station entry/exit counts, OD-level flows from the NUMBAT dataset, and train frequency data. This study builds an integrated view of demand, supply and crowding across Underground, Overground, DLR, Elizabeth Line, and Trams. It addresses three core questions:

1. How has demand changed from pre-COVID to post-COVID periods?
2. Which stations and origin-destination (OD) links are persistently crowded?
3. Can we predict which origin–destination (OD) links will be critically crowded in the near term?

The project conducts comprehensive exploratory data analysis (EDA), including temporal demand patterns, imbalance metrics, etc, which summarise the shape of demand by line, period, day, and year. The network-wide visualisations using chord diagrams for different time periods of a day. Additionally, the clustering analysis groups stations by time-of-day flow patterns and OD links by demand/crowding level. Finally, a machine learning model (XGBoost) is developed to predict crowding risk using engineered features capturing structural and temporal change. The model achieves a relatively high AUC (0.98) on held-out data and uses SHAP values for feature attribution. The entire pipeline is implemented in modular R scripts with reproducible outputs. The aims of such pipeline are:

- Load, clean, and standardise all raw spreadsheets;
- Apply TfL colour-coding conventions;
- Derive metrics such as passengers-per-train and station flow.

Visualisations, clustering, and modelling results are presented in detail in the full report `Summary_report.md`

2 Assessment Criteria

Technical Competence:

- Automated ingestion, Cleaned, joined, and analysed multi-source TfL datasets (NUMBAT, Footfall, Annual Counts, etc) into tidy formats using dplyr, tidyr packages and regex-based helpers.
- Included comprehensive Exploratory data analysis, network visualisation and clustering. In addition, built scalable features and trained an XGBoost crowding-alert model with 5-fold stratified CV, early stopping and SHAP interpretability.

User Interface:

- Clear folder and script structure with modular R files for reproducibility.
- Applied an official TfL colour palette to ensure consistency with actual Line.
- Employed embedded ggplot2 graphics, chord diagrams and interactive tooltips for exploratory analysis and network visualisation.

Analysis and Interpretation:

- It includes strong statistical summaries of pre/post-COVID demand trends and crowding, Link-period matrices, busiest-station league tables and imbalance charts to translate insights into actionable operational findings.
- Regarding modelling and prediction, it includes the interpretation of clustering patterns and model outputs (SHAP, ROC, calibration). SHAP analysis reveals that historical load factors outperform raw passenger growth as predictors.

Presentation and Communication:

- Designed concise summaries and readable visualisations with meaningful captions and titles to communicate key findings clearly. Detailed README and summary report with hyperlinks to all output figures and scripts.
- Created clean, consistent visuals (e.g., custom TfL colours) to enhance interpretability through intuitive layouts and descriptive figure captions.

Reproducibility and Documentation:

- Public GitHub repository includes all code, data, and Step-by-step run instructions in README, which enables full reproducibility of analysis and results. The README file includes all details for reproducibility.
- Raw and processed filtered datasets are clearly structured, and required packages are listed in a separate file.

Project Management:

- Clear separation of raw data, filtered outputs, code, and final outputs.
- Employed Git for version control, with regular commits documenting progressive changes, and used GitHub for remote repository management. And a final tagged release was archived as a zip for submission.

3 Project Reflection

Learnings:

- Learnt to design a modular, functional pipeline that memoises intermediate results, which significantly improved efficiency by avoiding redundant computation during repeated debugging and testing cycles.
- Implementing unit tests for data loaders revealed subtle errors, such as silent header shifts in 2023 NUMBAT files, that would have otherwise led to incorrect aggregations downstream. Therefore the preprocess data is crucial.
- Developed skills in building interpretable machine learning models (XGBoost + SHAP). Using SHAP values helped bridge the gap between machine learning model outputs and real-world interpretability.

Challenges:

- NUMBAT dataset required substantial cleaning and restructuring due to varying sheet formats. And it takes a long time to harmonise station names across the NUMBAT and Footfall datasets. And some of the data classification by TFL has changed since 2022 (e.g., TWT and MTT), which causes some challenges for comparison.
- The NUMBAT dataset varied in column names, sheet structures, and available metrics across years, requiring careful standardisation and custom loading logic.
- Designing a reliable predictive model required thoughtful feature engineering across time and structure. It always takes longer time to debug and cross-check multiple sub R scripts.

Further Development:

- It is worth exploring extending the current binary crowding-alert model into an ordinal classification framework that reflects multiple levels of crowding severity. Additionally, it can have experimented with temporal models, such as recurrent neural networks (RNNs) to capture longitudinal patterns across yearly OD flow deltas.
- In addition, it is interesting to use GNNs (Graph Neural Networks) to better model spatiotemporal correlations between stations and lines, potentially outperforming tabular models in future work.