TTC Bus Delay Report

Agam Sanghera, Ashita Diwan, Cheng Zhang, Yichun Liu

2024-12-16

Table of contents

Summary

This report examines the 2024 TTC bus delay dataset to forecast delay times based on factors such as route, incident type, location, and time. We aim to classify delays into short, medium, and long categories using a logistic regression model. The final logistic regression model show moderate effectiveness in predicting delay durations. Predicted frequencies for short and medium delays correspond with actual data; however, the model underpredicts long delays, highlighting the complexity of accurately capturing extended durations and their contributing factors. This study acts as a foundation for the implementation of real-time prediction models that could aid the Toronto Transit Commission in resource allocation and improving schedule adherence.

Introduction

Public transportation systems, such as Toronto's TTC, are essential for facilitating commuter mobility. However, delays are unavoidable and can affect the efficiency of services. Anticipating these delays may enhance operational decision-making and increase commuter satisfaction. The objective of this analysis is to identify the primary factors contributing to delays and to accurately forecast the duration of these delays by utilizing route, incident types, location, and time-related features as predictors.

Data

The data for this analysis was sourced from the open.toronto.ca website, with a specific emphasis on the bus delay data for the year 2024. Raw data can be found here.

Table 1: Snippet of TTC bus delay data

	Date	Route	Time	Day	Location	Incident	Min Delay	Min
0	2024-01-01	89	02:08	Monday	KEELE AND GLENLAKE	Vision	10	20
1	2024-01-01	39	02:30	Monday	FINCH STATION	General Delay	20	40
2	2024-01-01	300	03:13	Monday	BLOOR AND MANNING	General Delay	0	0
3	2024-01-01	65	03:23	Monday	PARLIAMENT AND BLOOR	Security	0	0
4	2024-01-01	113	03:37	Monday	MAIN STATION	Security	0	0

The dataset contains in total 45300 rows and 10 columns. Each row in the dataset represents one instance of delay, specifying its route, date and time, location, classification of bus incident, and its delay duration. A snippet of the dataset is show in Table 1.

Analysis

An EDA analysis is first conducted on the dataset with the folliowing objectives:

1. **Loading and Preprocessing Data:** Handling missing values, converting timestamp data to day parts, and cleaning data fields irrelevant to our delay analysis.
2. **Visualization:** Analyze the distribution of delays, identify top routes and locations with frequent delay incidents, and visualize delays based on day and incident type.

A linear regression model is then used to build the classification model to predict whether a delay falls into the short, medium or long duration. The C parameter in the linear regression model is chosen using a 5-fold cross validation with the classification accuracy as the metric.

The Python programming language (Van Rossum and Drake (2009)) and the following Python packages are used to perform tbe analysis: numpy (Harris et al. (2020)), Pandas (McKinney (2010)), altair (VanderPlas (2018)), scikit-learn (Pedregosa et al. (2011)). The code used to perform the analysis and generate the figures can be found here.

Results and Discussion

The EDA analysis of the TTC bus delay data uncovers several key insights.

Figure 1 shows that majority of the delays occur during the late evenings, most likely due to the influx of people returning back home from work.

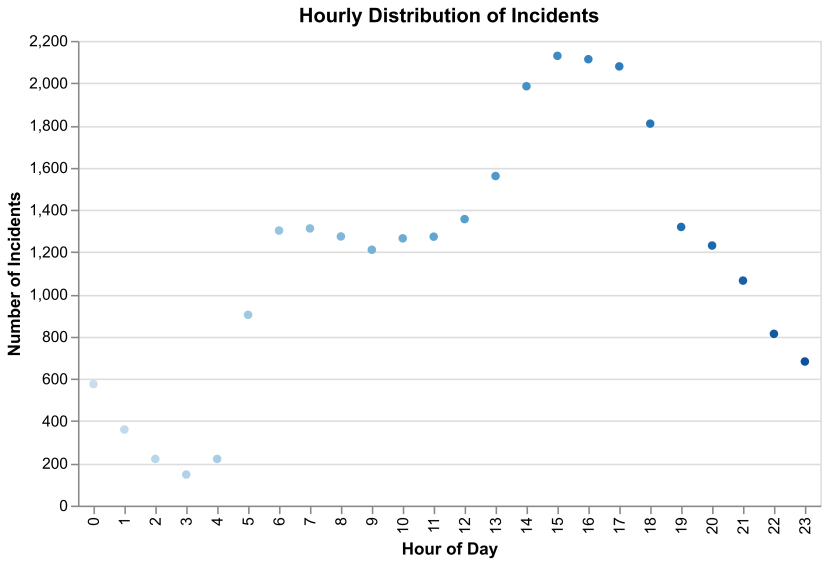

Figure 1

Figure 2 indicates that Tuesday through Friday are the days with the highest delays, suggesting a potential correlation with weekday commuter traffic.

Figure 3 show that mechanical issues are the primary cause of delays, comprising a substantial portion, followed by operator-related operations and diversions. This finding indicates potential areas for intervention, such as improved maintenance or optimized scheduling, to mitigate delay incidents.

The EDA analysis was very informative in understanding the columns of interest for this project, which will be used to create the logistic regression model to predict the expected delay. The delay output will be categorized into "Short", "Medium" or "Long". 5-fold cross-validation and randomized grid search were applied for C hyperparameter tuning to enhance model performance.

The results of the logistic regression model in Figure 4 show moderate effectiveness in predicting delay durations. Predicted frequencies for short and medium delays correspond with actual data; however, the model underpredicts long delays, highlighting the complexity of accurately capturing extended durations and their contributing factors. We could explore more advanced predictive models to improve accuracy. Furthermore, more data integration such as weather conditions could enhance model performance.

Harris, Charles R, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. 2020. "Array programming with NumPy." *Nature* 585 (7825): 357–62. https://doi.org/10.1038/s41586-020-2649-2.

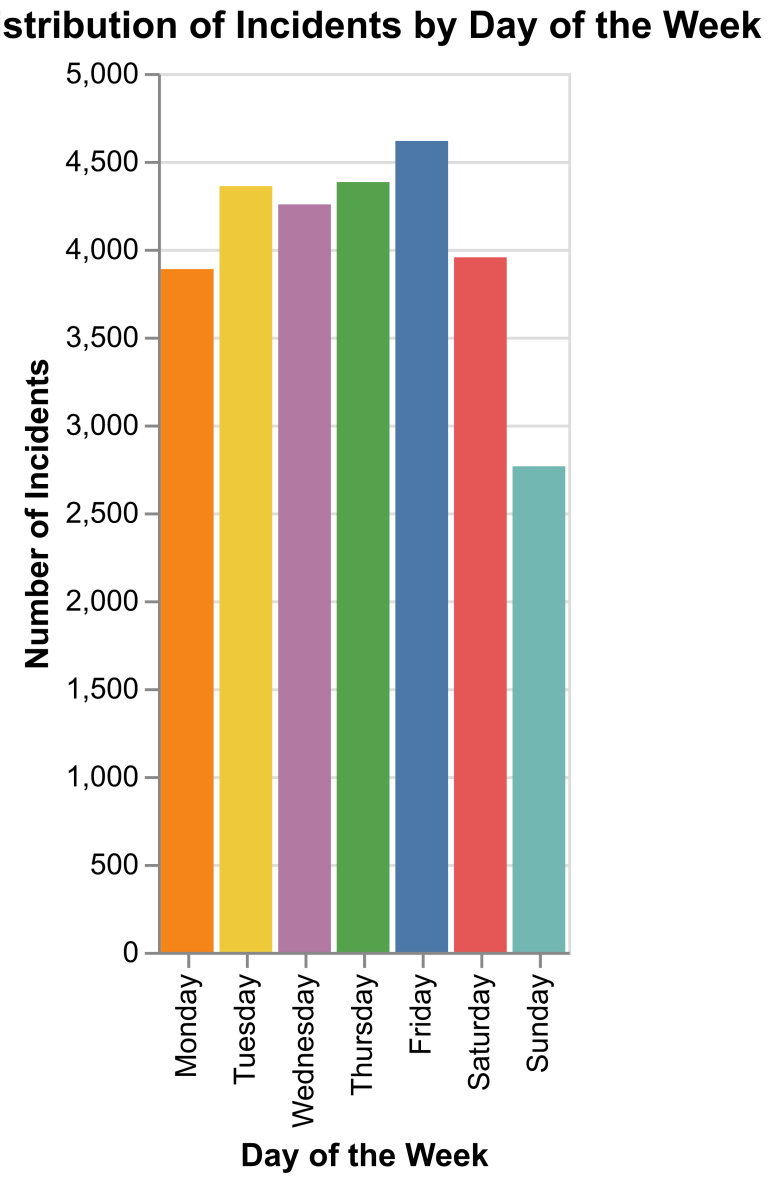
Distribution of Incidents by Day of the Week
Number of Incidents
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the Week

Figure 2

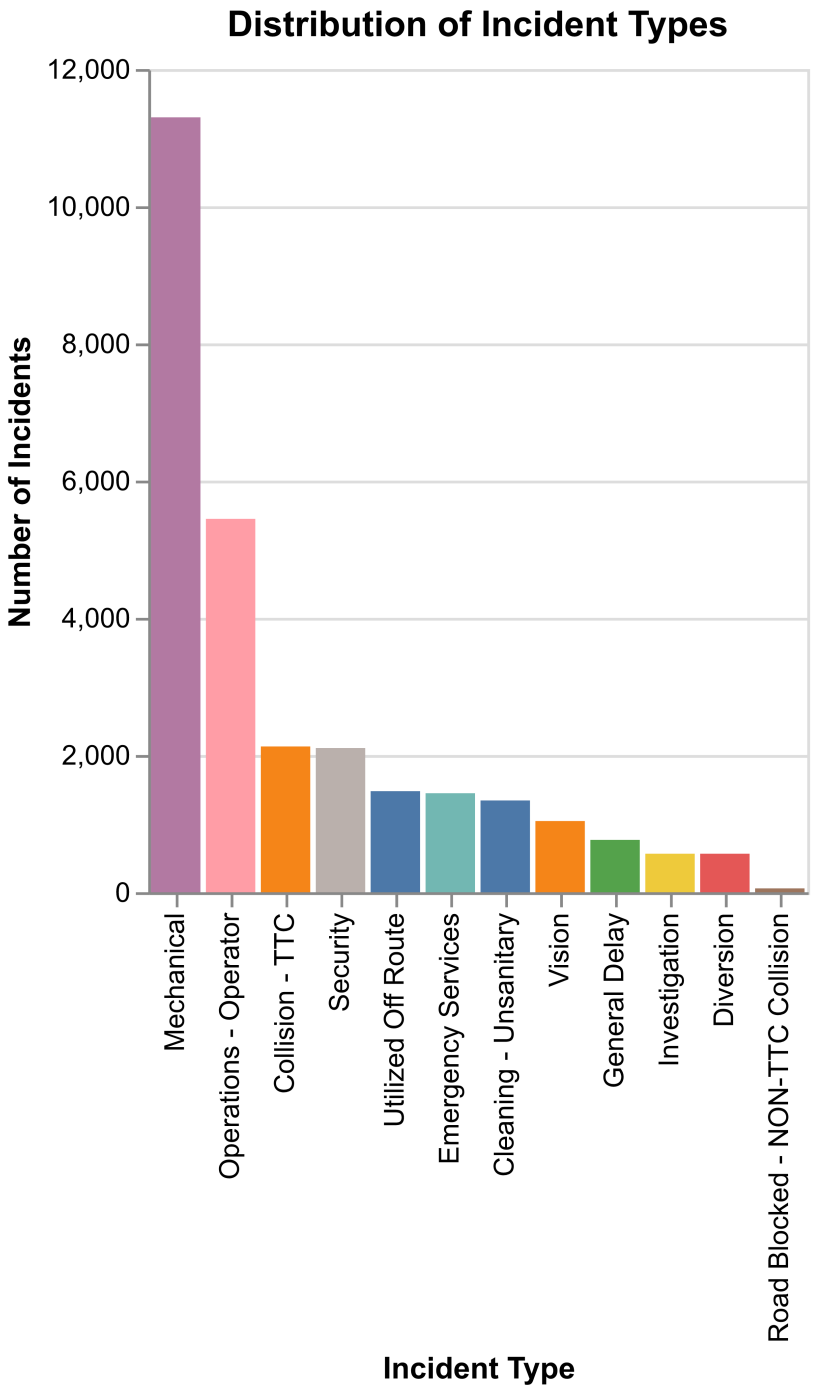
Distribution of Incident Types
Number of Incidents
12,000
10,000
8,000
6,000
4,000
2,000
0
Mechanical
Operations - Operator
Collision - TTC
Security
Utilized Off Route
Emergency Services
Cleaning - Unsanitary
Vision
General Delay
Investigation
Diversion
Road Blocked - NON-TTC Collision
Incident Type

Figure 3

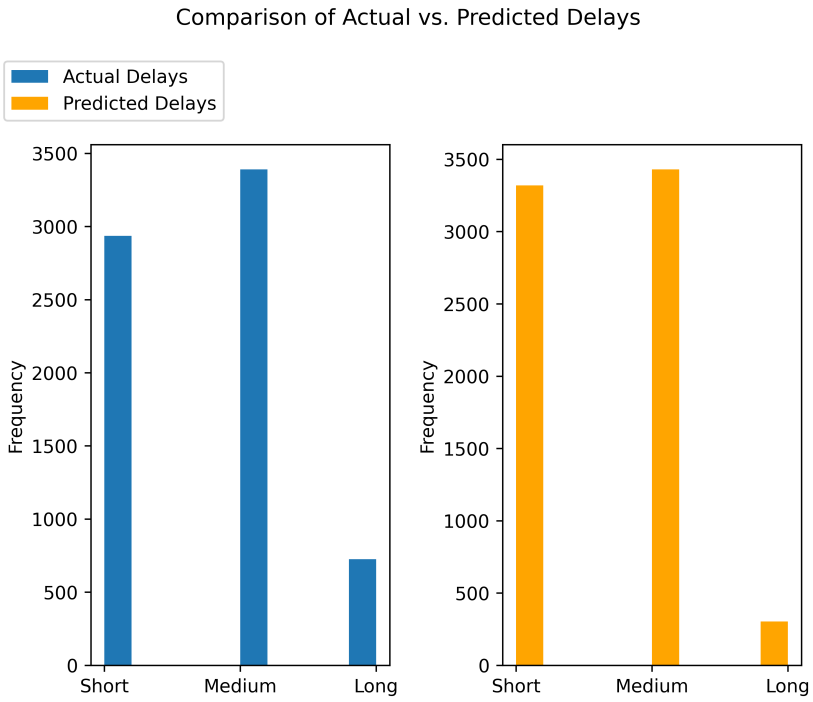
Comparison of Actual vs. Predicted Delays
Actual Delays
Predicted Delays
Frequency
3500
3000
2500
2000
1500
1000
500
0
Short
Medium
Long
Frequency
3500
3000
2500
2000
1500
1000
500
0
Short
Medium
Long

Figure 4

McKinney, Wes. 2010. "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, =51–56.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

VanderPlas, Jake. 2018. "Altair: Interactive Statistical Visualizations for Python." *Journal of Open Source Software* 3 (7825, 32): 1057. https://doi.org/10.21105/joss.01057.