NYC Taxi vs Citi Bike Wait- and Travel-Time Estimation

ECE 225A – Project
Atharv Nair (A69042035), Nitin Shreyes (A69041917)

December 14, 2025

Abstract

We build a reproducible pipeline for comparing New York City taxi and Citi Bike service quality. The first pillar estimates rider wait times by modeling quarter-hour arrivals (split by rush/off-peak and weekday/weekend) as Poisson / Negative-Binomial processes and translating inter-arrival times into exponential wait-time distributions. The second pillar focuses on travel-time estimation: we fit discrete Gamma distributions per distance cohort and a continuous lognormal accelerated failure-time model—augmented with an e-bike indicator—that removes the need for coarse bins. Both tracks feed a Streamlit decision-support dashboard backed by cached Parquet/JSON artifacts. This report describes the data, modeling choices, UX considerations, and future work.

1 Introduction

When speed is the only objective a yellow cab usually wins, but cost, emissions, and congestion make Citi Bike attractive if the time penalty is small. Our goal is to quantify *how much slower biking really is* across neighborhoods and times of day. Rather than mimic a full routing engine we zoom in on the probability distributions that control two door-to-door components:

- **Wait time.** How long until a taxi arrives or a bike becomes available in the nearby dock cluster?
- **Travel time.** Once a rider is moving, how long will the trip take as a function of distance and traffic patterns?

To answer these questions we fit probability distributions directly to the NYC Taxi and Citi Bike datasets (Jan–Jun 2024). We analyze Poisson and Negative-Binomial arrival counts for diagnostic purposes, but the deployed wait estimator ultimately uses the implied exponential mean because it matches the observed inter-arrival histogram. Likewise, we study Gamma cohorts to understand distance-dependent variance, but the app relies on a lognormal accelerated failure-time regression for travel minutes because it provides smooth predictions without bin boundaries. Both estimates feed a public Streamlit app (https://nycpublictransit.streamlit.app/) that lets users click origin/destination pairs and instantly compare modes (Fig. 1).

Assumptions and design choices:

figures/website.png

Figure 1: Public dashboard (https://nycpublictransit.streamlit.app/) showing taxi vs Citi Bike recommendations for arbitrary O/D pairs.

- Wait times rely on inter-arrival gaps as a proxy for rider experience. Stations within 200 m are pooled into "catchments" to reflect a rider's willingness to walk to the next dock.
- Travel-time samples are restricted to 1–120 minutes and within 12 km to keep Citi Bike and taxi cohorts comparable. Forecasts outside this range fall back to coarser heuristics.
- We only use features available historically (distance, rush/weekend indicators, e-bike/classic flag). Weather, incidents, dynamic pricing, and subway usage are left for future work; adding subway headways is the next obvious extension once this pipeline is stable.

2 Dataset

Our pipelines ingest the official NYC TLC yellow taxi trip records [1] and the Citi Bike historical CSVs [2] for the first half of 2024. Both sources provide raw trip-level logs

rather than aggregates, which allows us to recompute arrivals, wait gaps, and travel durations with consistent filters.

Yellow taxi. Six Parquet files ('yellow_tripdata_2024-01' through '-06') contribute 17.6 million filtered trips after removing zero-distance or >120 minute rides. Each record contains pickup/dropoff timestamps, trip distance (miles), and TLC pickup/dropoff zone IDs (263 unique zones citywide). These columns let us (i) count hourly arrivals per zone for the wait-time analysis, and (ii) compute true travel minutes and straight-line distances for the lognormal regression. Taxi fares, passenger counts, and payment types are present in the raw files but remain unused in this iteration.

Citi Bike. The Jan–Jun 2024 CSVs add 18.3 million rides spanning 2,223 unique start stations and 2,240 end stations. Each row includes start/end timestamps, station coordinates, rideable type (classic vs electric), and membership flag. Citi Bike coverage is highly concentrated in Manhattan and northwest Brooklyn; more than half of the rides originate in Manhattan and ≈ 450,000 trips begin and end at the same dock (short rebalancing hops). Because riders will often walk to nearby docks, we cluster stations within 200 m into catchments and track arrival counts at that level for the wait-time model.

Table 1 summarizes the dataset scales used in the deployed models. Subway feeds and weather data are not yet integrated; adding them is the next major item after stabilizing the taxi/bike comparisons.

3 Wait-Time Estimation

Our starting point was the textbook assumption that arrivals follow a Poisson process, meaning the hourly count N_t for a zone or station has mean and variance both equal to λ_h. This worked in the busiest taxi zones but failed for bikes: weather swings or rebalancing trucks often sent the variance far above the mean. We therefore relaxed the model to a Negative-Binomial, which keeps the same mean λ_h but adds a dispersion term so that quieter locations are no longer forced to look Poisson. The NB fit is purely about understanding how many pickups happen in each hour.

Once an arrival rate is available we need an approximate wait time. We cannot observe actual rider waits, so we use the *inter-arrival gap*—the time difference between two consecutive pickups—as a proxy. This is imperfect: it assumes riders show up uniformly at random and that bikes/taxis are immediately available once a previous customer leaves. In reality someone could arrive during a lull or a bike dock might be empty even though the last checkout just happened. Still, the proxy aligns well with intuition and gives us a measurable quantity.

Plotting the inter-arrival gaps revealed a clear exponential shape for both taxis and bikes (after pooling nearby docks into catchment areas). Figure 2 shows two representative cohorts in the *top row*: the empirical histogram (blue) decays at roughly the same rate as the exponential curve (green), while the Poisson and Negative-Binomial fits describe the arrival counts themselves. Because the exponential mean equals $1/\lambda$ we can turn each hourly arrival rate into a wait-time estimate with a single formula. These exponentials become the default wait model inside the Streamlit app, while the NB parameters remain in the cache for diagnostics and future improvements.

4 Travel-Time Estimation

We start by looking at simple **Gamma cohorts**. Trips are grouped by mode, 2 km distance buckets, and rush/weekend flags; the mean and variance within each group define a Gamma curve. These summaries act as sanity checks (e.g., bikes at 4–6 km rush hour look much more spread out than taxis) and as fallbacks whenever more advanced models lack data.

Because we want smooth predictions for any distance, we train one **lognormal regression** over all trips. Let T denote travel minutes for a trip with distance d. We model

$$\log T = \beta_0 + \beta_1 d + \beta_2 d^2 + \beta_3 \mathbb{1}_{\text{rush}} + \beta_4 \mathbb{1}_{\text{weekend}} + \beta_5 \mathbb{1}_{\text{ebike}} + \varepsilon,$$

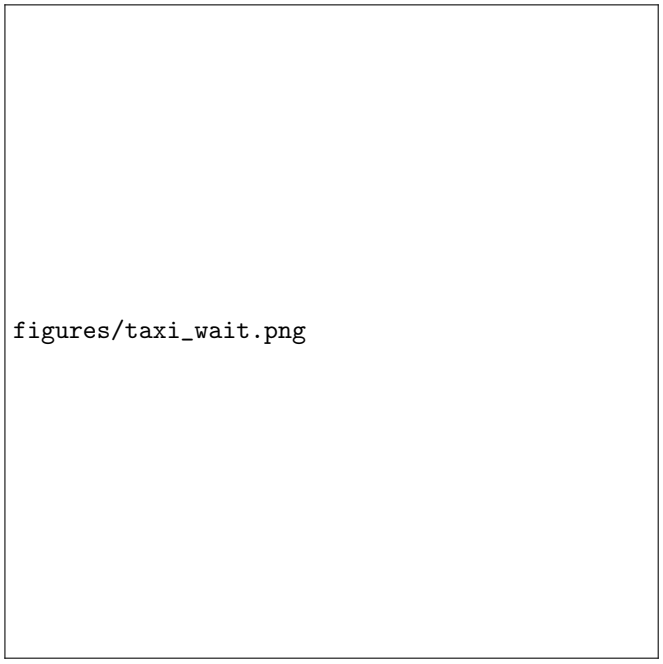
where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and the e-bike indicator is set to zero for taxis. We fit this Gaussian GLM using the maximum-likelihood solver in `statsmodels.api.GLM`, which is equivalent to least squares on $\log T$. Coefficients therefore correspond to percentage changes in expected time: multiplying d by one kilometer adjusts the log mean by β_1, so the minutes scale by e^{β_1}. Table 2 shows that e-bikes reduce travel time by roughly 18% ($e^{-0.193}$), while rush-hour increases taxi time by about 6% ($e^{0.059}$).

To make predictions we plug a new trip's features into the fitted linear equation, obtain $\hat{\mu} = X\hat{\beta}$, and convert back to minutes with the lognormal mean formula $\hat{T} = \exp(\hat{\mu} + 0.5\hat{\sigma}^2)$. This correction term accounts for the asymmetry introduced by exponentiation. The approach behaves smoothly across distances yet still mirrors the Gamma diagnostics discussed earlier.

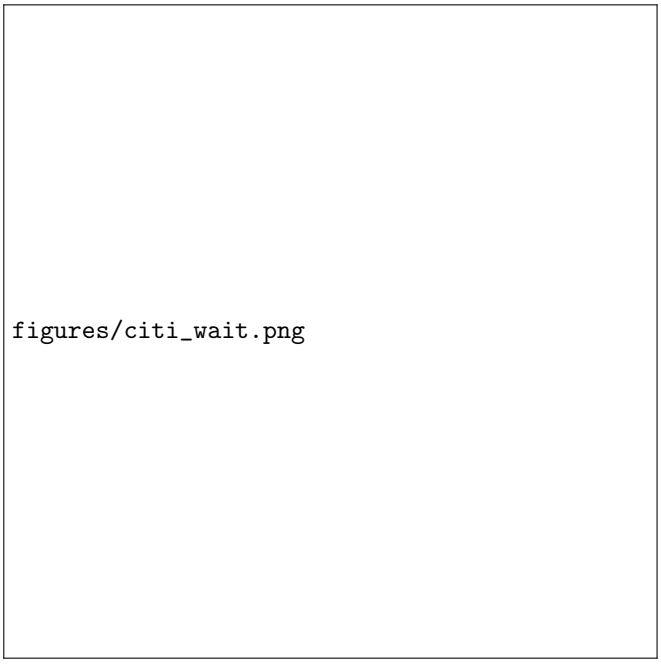
Quality metrics (log-space $R^2 \approx 0.62$ and MAE≈ 0.33 for bikes) indicate the simple feature set explains most of the variance; rush/weekend dummies only add small adjustments once distance is accounted for. If the regression cannot be evaluated—for example, a request beyond the 12 km cap—we fall back to the Gamma cohort average and, as a last resort, to a constant-speed heuristic. The bottom row of Fig. 2 shows how the lognormal fit smooths the Gamma histogram without losing the overall shape for both taxis and bikes.

5 Streamlit Dashboard

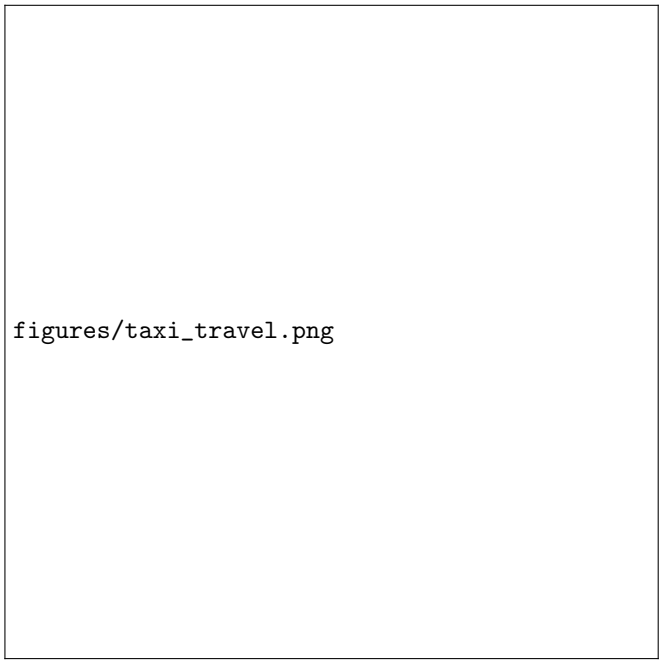
The front end (https://github.com/AtharvRN/NYC_Public_Transit) uses Streamlit to let users:



Taxi wait: zone 3905.15, weekday off-peak



Citi Bike wait: Financial District, weekend rush



Taxi travel: 0–2 km, rush-hour weekday

Citi Bike travel: 0–2 km, rush-hour weekday

Figure 2: Model diagnostics spanning the two modeling tasks: top row shows arrival/wait fits (Section 3), bottom row shows travel-time Gamma vs lognormal fits (Section 4).

Table 1: Key statistics for the Jan–Jun 2024 training data.

Metric	Taxi	Citi Bike
Trips used in travel-time model	17,550,334	18,253,964
Unique pickup locations	263 TLC zones	2,336 stations (1,575 catchmen
Months covered	Jan–Jun 2024	Jan–Jun 2024
Metadata	Distance, zone IDs	Station coords, rideable type
Active wait cohorts (location/hour/rush/weekend)	5,975	47,876

Table 2: Lognormal GLM parameters extracted from `travel_lognormal_glm.json`.

Mode	σ	β_d	β_{d^2}	β_{rush}	β_{ebike}
Taxi	0.391	0.499	−0.031	0.059	0.000
Bike	0.485	0.702	−0.049	−0.025	−0.193

- Pick origin/destination on an interactive map (Folium).
- View taxi vs bike wait times for the selected locations, with automatic fallbacks to the nearest station if the exact cohort is missing.
- See travel-time estimates using the lognormal GLM (with Gamma fallback).
- Compare total journey time distributions.

5.1 UX Notes

Tabs separate taxi/bike diagnostics; tooltips explain why certain cohorts are disabled (insufficient samples). We plan to add UI screenshots here (placeholder Fig. 3).

6 Limitations & Future Work

- **Modeling assumptions:** Wait-times use inter-arrival proxies; no weather or traffic features; GLM only uses distance/rush/weekend/e-bike.
- **Distance calculation:** Haversine distance does not account to the roads and turns involved in reaching the end location
- **Data coverage:** Only Jan–Jun 2024; Citi Bike GPS routes or real-time inventory (docks per station) are not modeled.
- **Costs:** Trip fares are not surfaced; taxi Parquet files contain the needed columns but Citi Bike pricing must be inferred from membership tiers.
- **Scalability:** Subway data is not integrated yet—downloading GTFS and MTA ridership feeds is planned but time-consuming.
- **Future features:** Add uncertainty bands, enable user-uploaded O/D pairs, integrate subway travel-time estimators, and experiment with probabilistic travel-time percentiles.

7 Discussion

Speed modelling was the other approach used to estimate travel time. Similar to earlier methods, the data were grouped according to rush hours and weekends. However, due to the limited number of entries for many unique routes, this grouping scheme was not able to properly distinguish between different routes. Furthermore, across all four categorizations, the observations were heavily concentrated around the mean, leading to underdispersion. As a result, Poisson and negative binomial distributions were not suitable for capturing the underlying variability in travel times. Instead, a Weibull distribution was adopted, as it provided a higher AIC score compared to the gamma and lognormal alternatives.

In contrast to the taxi dataset, not all entries in the Citi Bike data contributed meaningfully to modelling New York's traffic conditions. More than 450,000 entries had identical start and end stations. Many of these trips were concentrated around peak tourist locations, such as Central Park 6 Ave, where over 12% of rides began and ended at the same station. While such usage patterns are reasonable in practice—for example, casual tourist rides that start and end at the same landmark—they reduce the effective amount of information available for our traffic-focused modelling, since the perceived volume of data is much larger than the subset that is actually informative for our purposes.

Arrival-count modelling for both datasets also did not conform well to a Poisson distribution. A likely explanation is the dependence of arrival counts on time-varying and context-specific factors such as time of day, day of the week, public holidays, and city events. These factors introduce strong temporal patterns and clustering in arrivals, violating the Poisson assumptions of independence and a constant arrival rate over the counting interval. While our datasets did not include explicit information on these finer-grained covariates, incorporating them would have required a more complex model and substantially more data than were available. For instance, in the Citi Bike dataset, although the total number of rows was large, the number of rides within a specific hour on a given weekday (e.g., a Wednesday) was in the single digits for many stations in cases, limiting our ability to reliably model hourly arrival patterns.

8 Conclusion

We delivered a reproducible workflow: raw data ingestion, exploratory notebooks, lognormal travel-time modeling with e-bike adjustments, cached wait-time summaries, and a Streamlit app that gracefully degrades to nearby stations. The lognormal GLM improves Citi Bike estimates substantially over Gamma bins while keeping the pipeline explainable. Next steps include richer covariates, explicit cost modeling, subway integration, and automated CI for the derived artifacts.

References

[1] NYC Taxi & Limousine Commission, "TLC Trip Record Data," https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[2] Citi Bike NYC, "Historical Trip Data," https://s3.amazonaws.com/tripdata/index.html