NYC Taxi Trip Duration Prediction: Exploratory Analysis and Regression Modelling

Munal Baraili
900006725

TAFE NSW

Contents

Introduction

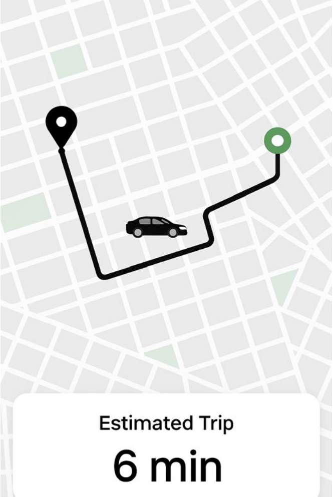
In this project, we analyse the NYC Taxi Trip Duration dataset, which contains detailed records of individual taxi rides, including vendor ID, passenger count, pickup and drop-off timestamps, GPS coordinates, and trip duration in seconds. The aim is to explore the data, understand the behaviour of key variables, and investigate how trip duration relates to spatial and temporal trip characteristics.

We begin with basic data cleaning to handle invalid or extreme values and to prepare the dataset for analysis. We then conduct exploratory data analysis (EDA) to examine the distributions of key features and to identify patterns in trip duration across distance, passenger count, weekday, and time of day.

Finally, we build and evaluate machine learning models to predict trip duration from the engineered features. A simple Linear Regression model on a log-transformed target serves as a baseline, and a Random Forest Regressor is used to capture nonlinear relationships. Model performance is measured on an unseen test split using RMSE, MAE, and R^2, complemented by visual diagnostics such as prediction–vs–actual plots and residual analysis. This report summarises the data preparation, EDA, modelling process, results, and the main insights obtained.

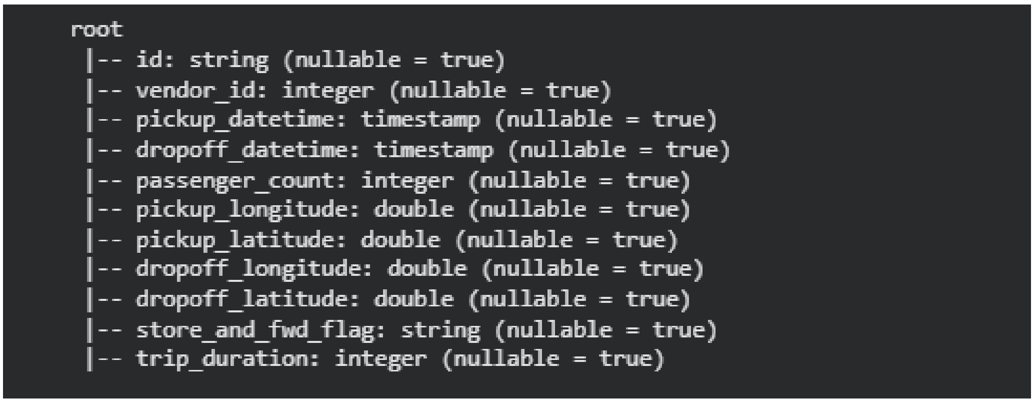
Objective

The objective of this project is to clean, explore, and model the NYC Taxi Trip dataset to understand what drives trip duration and to build a model that can accurately predict it. By analysing key features and their relationship with travel time, the project aims to extract useful insights and create a predictive system that performs well on unseen data. At a high level, the idea is simple: given a pickup point and destination, estimate how long the trip will take—just like telling your boss, partner, or friend when you'll arrive. While real travel time depends on many factors like traffic or weather, this project focuses on predicting a reliable baseline estimate.



Dataset Description and Initial Exploration

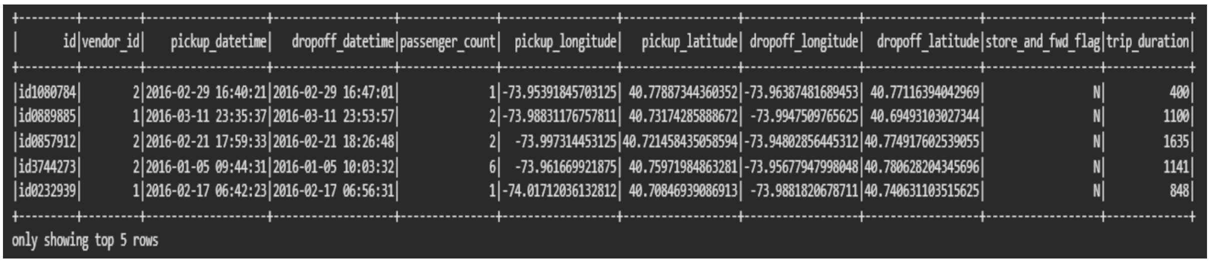
The dataset provided, *nyc_taxi_trip_duration.csv*, contains detailed records of individual taxi trips collected across New York City. Before applying any preprocessing, exploring the schema and the first few rows helps us understand the structure of the data and the meaning of each variable.



```
root
 |-- id: string (nullable = true)
 |-- vendor_id: integer (nullable = true)
 |-- pickup_datetime: timestamp (nullable = true)
 |-- dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- pickup_longitude: double (nullable = true)
 |-- pickup_latitude: double (nullable = true)
 |-- dropoff_longitude: double (nullable = true)
 |-- dropoff_latitude: double (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- trip_duration: integer (nullable = true)
```

schema of dataset

The dataset includes the following attributes:
id – A unique identifier assigned to each trip.
vendor_id – Indicates the taxi company (1 or 2). These reflect different data collection systems used by NYC taxi vendors.
pickup_datetime and dropoff_datetime – Timestamps in the format *YYYY-MM-DD HH:MM: SS*, marking when the trip began and ended.
passenger_count – Number of passengers recorded for the trip.
pickup_longitude, pickup_latitude – GPS coordinates for the pickup location.
dropoff_longitude, dropoff_latitude – GPS coordinates for the drop-off location.
store_and_fwd_flag – A system flag ("Y" or "N"). It indicates whether the trip record was held temporarily and forwarded later because of a connectivity issue.
trip_duration – The total travel time for the trip, recorded in seconds. This serves as the target variable for prediction.



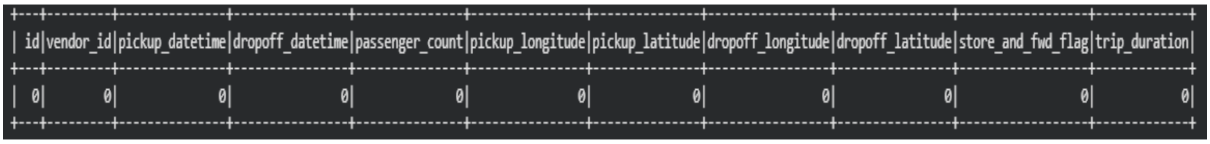
```
+---------+---------+-------------------+-------------------+---------------+------------------+------------------+------------------+------------------+------------------+-------------+
|       id|vendor_id|    pickup_datetime|   dropoff_datetime|passenger_count|  pickup_longitude|   pickup_latitude| dropoff_longitude|  dropoff_latitude|store_and_fwd_flag|trip_duration|
+---------+---------+-------------------+-------------------+---------------+------------------+------------------+------------------+------------------+------------------+-------------+
|id1080784|        2|2016-02-29 16:40:21|2016-02-29 16:47:01|              1|-73.95391845703125| 40.77887344360352|-73.96387481689453| 40.77116394042969|                 N|          400|
|id0889885|        1|2016-03-11 23:35:37|2016-03-11 23:53:57|              2|-73.98831176757811| 40.73174285888672| -73.9947509765625| 40.69493103027344|                 N|         1100|
|id0857912|        2|2016-02-21 17:59:33|2016-02-21 18:26:48|              2|  -73.997314453125|40.721458435058594|-73.94802856445312|40.774917602539055|                 N|         1635|
|id3744273|        2|2016-01-05 09:44:31|2016-01-05 10:03:32|              6|  -73.961669921875| 40.75971984863281|-73.95677947998048|40.780628204345696|                 N|         1141|
|id0232939|        1|2016-02-17 06:42:23|2016-02-17 06:56:31|              1|-74.01712036132812| 40.70846939086913| -73.9881820678711|40.740631103515625|                 N|          848|
+---------+---------+-------------------+-------------------+---------------+------------------+------------------+------------------+------------------+------------------+-------------+
only showing top 5 rows
```

Peeking top 5 rows

Trip duration is measured in seconds in the original NYC Taxi Trip Duration competition dataset, and this formulation is consistent here. Using seconds avoids ambiguity and makes modelling easier, especially because durations can vary from under a minute to several hours.

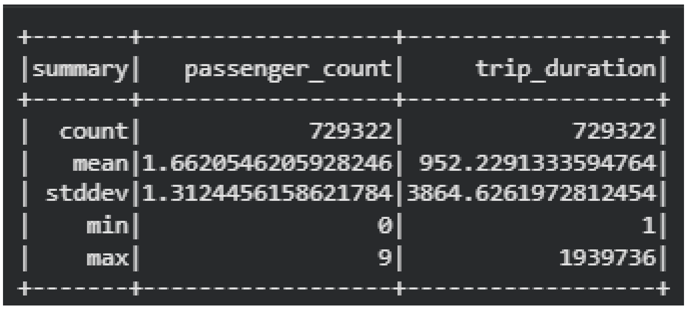
Data Cleaning and Preprocessing

Before applying any transformations, a quick structural check showed that all columns had consistent non-null counts. This suggests that the dataset is complete in terms of field availability, although this does not rule out invalid entries or extreme outliers.



```
+---+---------+---------------+----------------+---------------+----------------+---------------+-----------------+----------------+------------------+-------------+
| id|vendor_id|pickup_datetime|dropoff_datetime|passenger_count|pickup_longitude|pickup_latitude|dropoff_longitude|dropoff_latitude|store_and_fwd_flag|trip_duration|
+---+---------+---------------+----------------+---------------+----------------+---------------+-----------------+----------------+------------------+-------------+
|  0|        0|              0|               0|              0|               0|              0|                0|               0|                 0|            0|
+---+---------+---------------+----------------+---------------+----------------+---------------+-----------------+----------------+------------------+-------------+
```

A brief exploratory scan revealed several irregular patterns—for example, passenger counts recorded as 0, 7, or 9, and trip durations extending into multiple hours or even days. These issues require cleaning to ensure that the modelling process is not distorted by unrealistic observations.

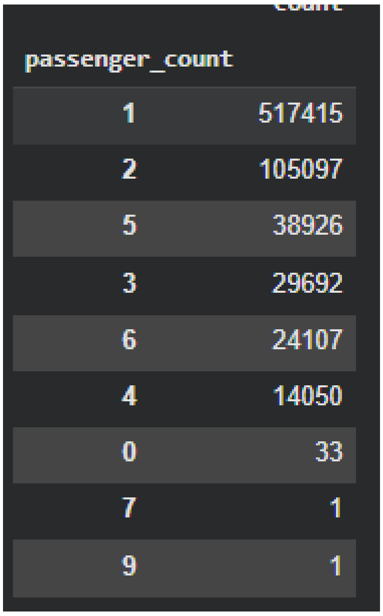


```
+-------+------------------+------------------+
|summary|   passenger_count|     trip_duration|
+-------+------------------+------------------+
|  count|            729322|            729322|
|   mean|1.6620546205928246| 952.2291333594764|
| stddev|1.3124456158621784|3864.6261972812454|
|    min|                 0|                 1|
|    max|                 9|           1939736|
+-------+------------------+------------------+
```

We also assume that the pickup and drop-off timestamps generally align with the recorded trip duration, although this assumption is verified more closely during preprocessing.

Handling Outliers

Passenger Count

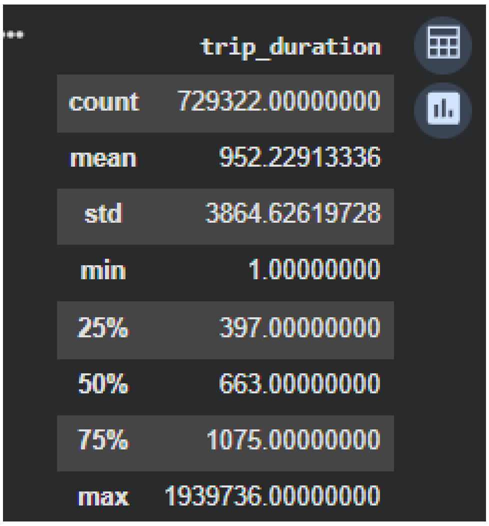


	count
passenger_count	
1	517415
2	105097
5	38926
3	29692
6	24107
4	14050
0	33
7	1
9	1

Most trips involve between 1 and 3 passengers, with a sharp concentration around 1. Values such as 0, 7, and 9 passengers occur extremely rarely and do not represent realistic taxi trips. To stabilise the feature without discarding rows, these values were replaced with 2, which is close to the empirical average passenger count (~1.66).

This preserves data volume while avoiding the influence of implausible or sensor-error records.

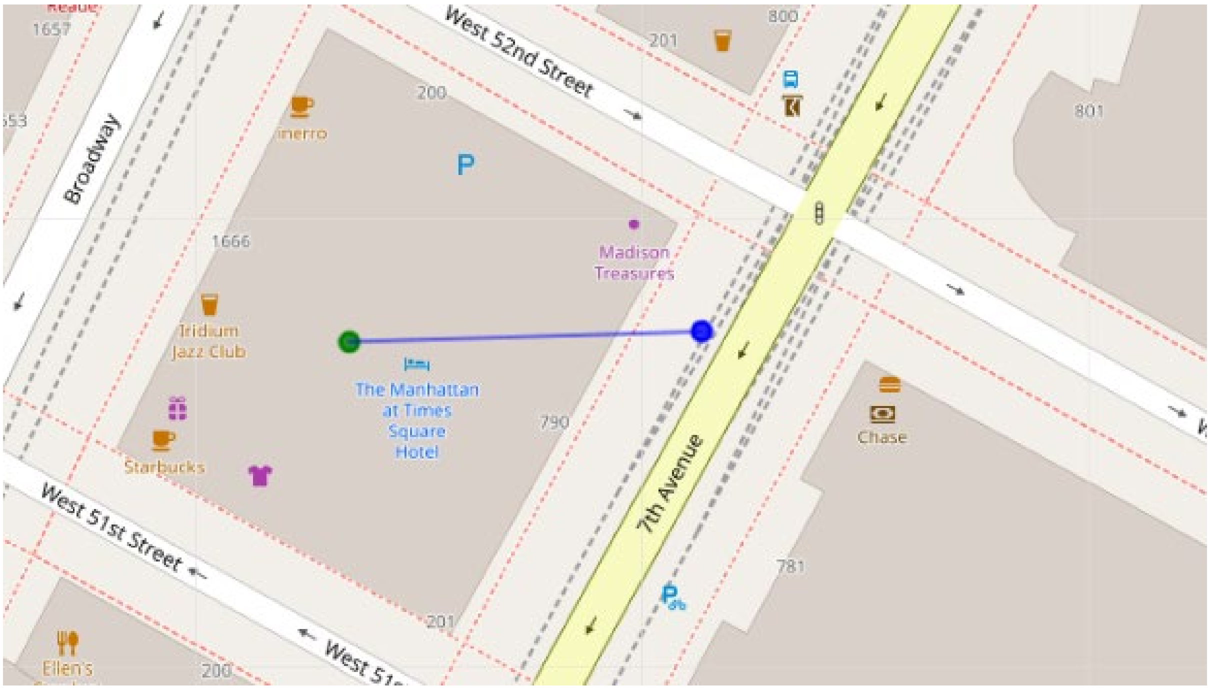
Trip Duration



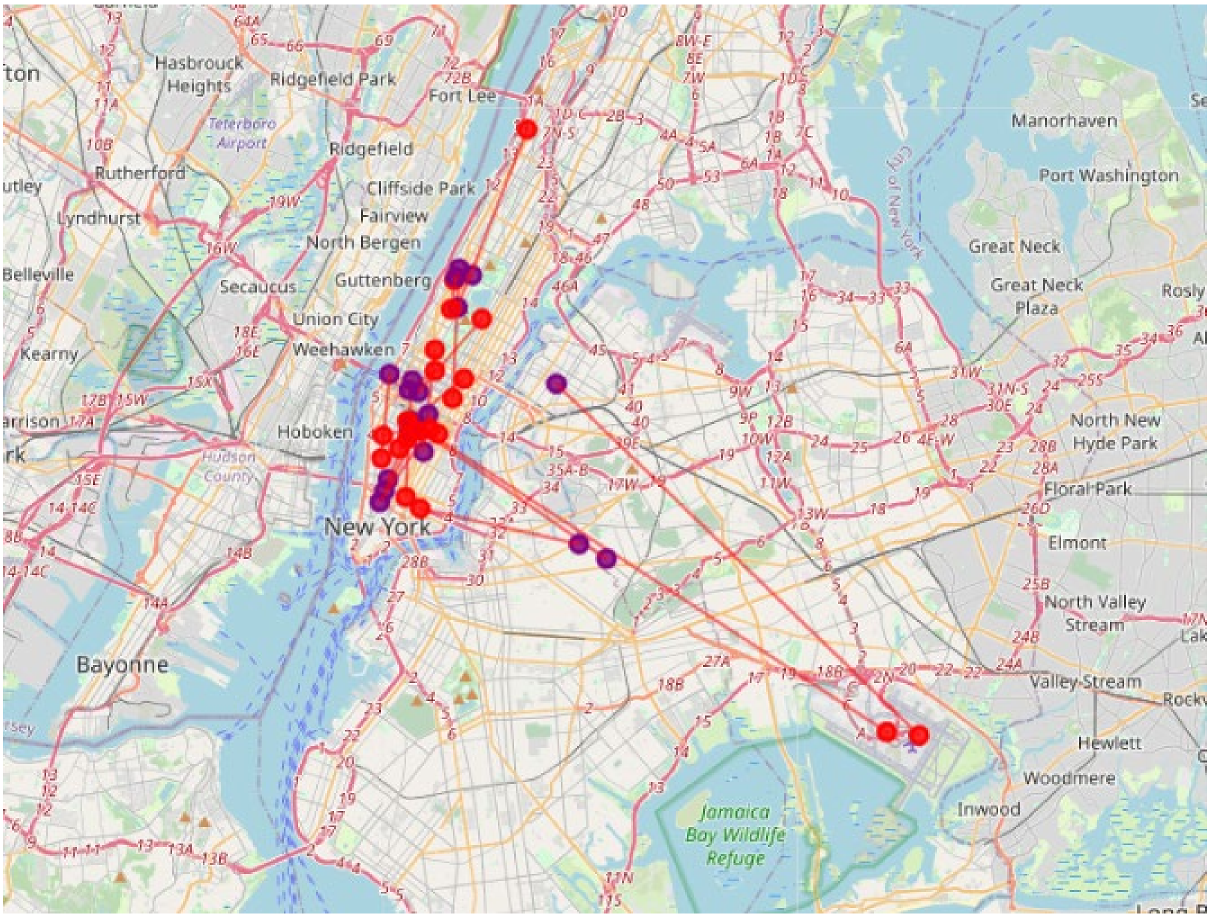
	trip_duration
count	729322.00000000
mean	952.22913336
std	3864.62619728
min	1.00000000
25%	397.00000000
50%	663.00000000
75%	1075.00000000
max	1939736.00000000

Trip duration is recorded in seconds, and its distribution is highly right skewed. Two kinds of problematic durations were observed:

- Trips shorter than 60 seconds



- Trips longer than 4 hours — approximately 1,015 cases, many appearing normal in terms of pickup and dropoff coordinates but implausible in terms of time spent.



Together, these account for 5,286 rows (0.72% of the dataset).
Because they cannot be reliably corrected and would heavily distort any regression model, they were removed completely.

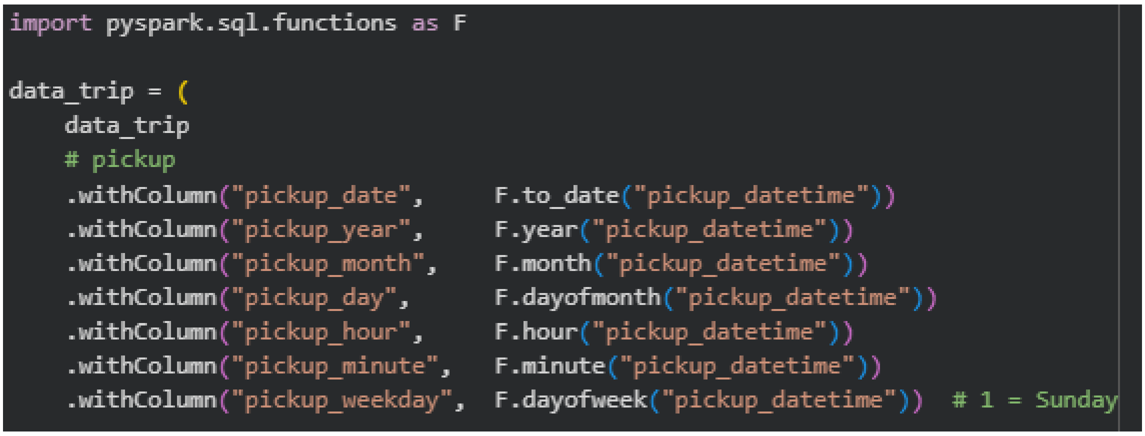
Extracting the Clean Dataset

After handling passenger-count anomalies and removing extreme trip-duration values, we extracted a clean version of the dataset for further feature engineering and modelling. The cleaned data retains the core structure and variability of NYC taxi behaviour while eliminating values that would bias model training.

Feature Engineering

1. Temporal Features from Pickup Datetime

These features allow the model to learn patterns such as morning congestion, weekend travel differences, and seasonal variation.



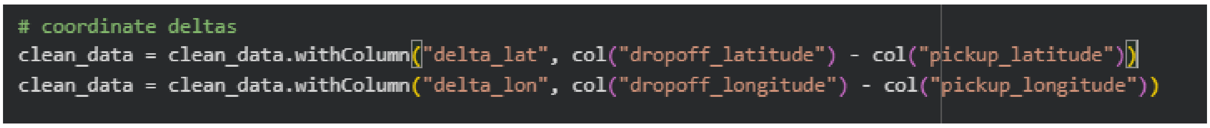
```
import pyspark.sql.functions as F

data_trip = (
    data_trip
    # pickup
    .withColumn("pickup_date",     F.to_date("pickup_datetime"))
    .withColumn("pickup_year",     F.year("pickup_datetime"))
    .withColumn("pickup_month",    F.month("pickup_datetime"))
    .withColumn("pickup_day",      F.dayofmonth("pickup_datetime"))
    .withColumn("pickup_hour",     F.hour("pickup_datetime"))
    .withColumn("pickup_minute",   F.minute("pickup_datetime"))
    .withColumn("pickup_weekday",  F.dayofweek("pickup_datetime"))  # 1 = Sunday
```

Taxi behaviour varies strongly by hour of day, day of week, and month. To capture these patterns, the following features were extracted from pickup_datetime:

- pickup_hour – 0–23
- pickup_weekday – 0 (Monday) to 6 (Sunday)
- pickup_month – 1–12

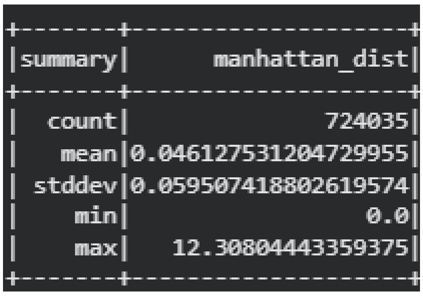
2. Spatial Direction Features



```
# coordinate deltas
clean_data = clean_data.withColumn("delta_lat", col("dropoff_latitude") - col("pickup_latitude"))
clean_data = clean_data.withColumn("delta_lon", col("dropoff_longitude") - col("pickup_longitude"))
```

To represent the directional shift between pickup and dropoff points, two difference-based features were created.

3. Manhattan Distance



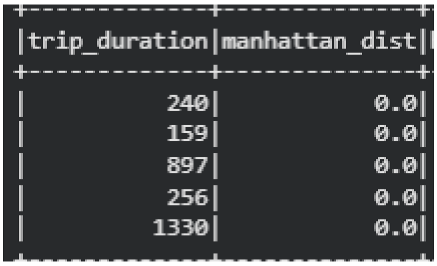
```
+-------+--------------------+
|summary|      manhattan_dist|
+-------+--------------------+
|  count|              724035|
|   mean|0.046127531204729955|
| stddev|0.059507418802619574|
|    min|                 0.0|
|    max|   12.30804443359375|
+-------+--------------------+
```

Because NYC’s Street network is largely grid-shaped, we computed Manhattan Distance (L1 distance) as:

manhattan_dist = |delta_lat| + |delta_lon|

It approximates how far a vehicle would realistically travel when restricted to rectangular street paths.

4. Zero-Distance Anomalies



```
+-------------+--------------+
|trip_duration|manhattan_dist|
+-------------+--------------+
|          240|           0.0|
|          159|           0.0|
|          897|           0.0|
|          256|           0.0|
|         1330|           0.0|
+-------------+--------------+
```

During feature creation, several rows showed:

- manhattan_dist = 0

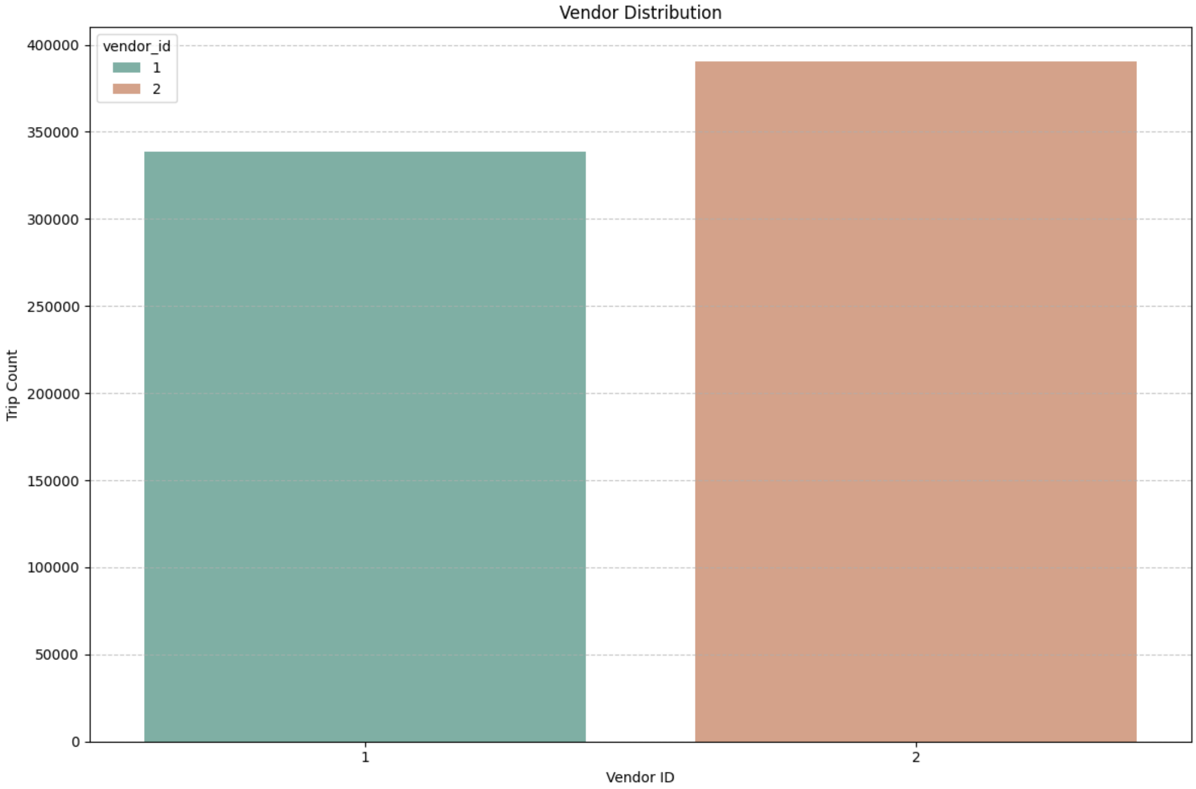
yet had non-zero trip_duration, sometimes lasting several minutes.

These rows were later removed during cleaning, as they do not provide meaningful spatial information and would mislead the model.

Exploratory Data Analysis (Univariate & Bivariate)

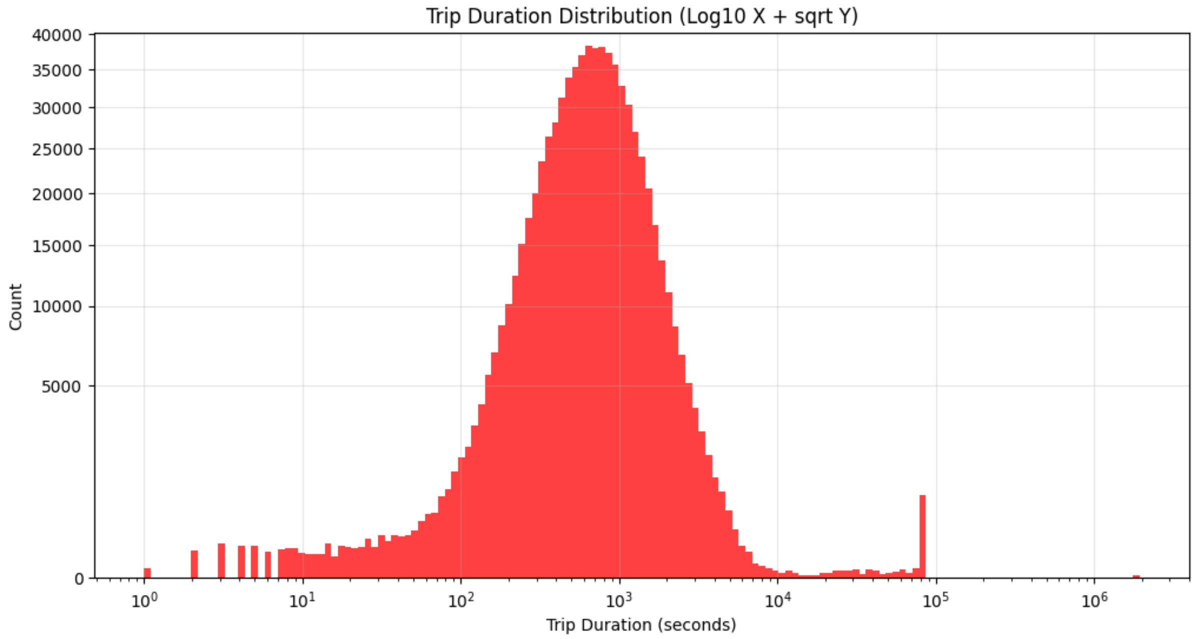
To understand the behaviour of the dataset before modelling, we first look at how each variable is distributed on its own, and then how trip duration changes across different groups such as vendor, passenger count, time, and location.

Vendor Distribution to Trip Count



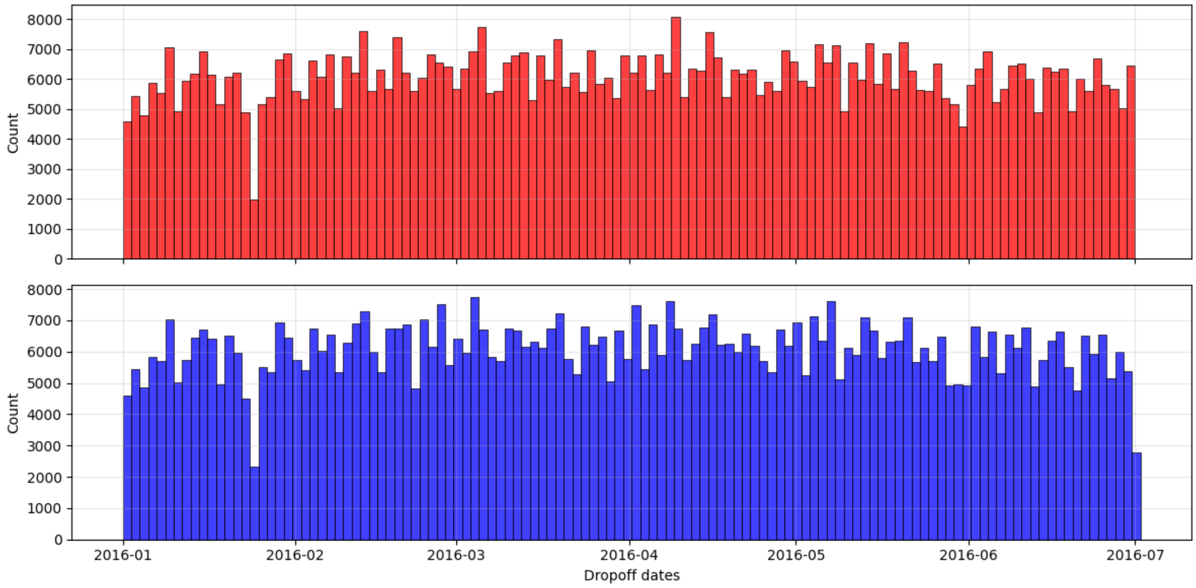
For vendor ID, the dataset is roughly balanced between Vendor 1 and Vendor 2, with Vendor 2 having slightly more trips overall. Passenger count is heavily concentrated between 1 and 3 passengers, with single-passenger trips dominating. Values like 0, 7, and 9 passengers appear only a few times and already look suspicious at this stage, which later motivates treating them as outliers in the cleaning phase.

Trip Duration Distribution.

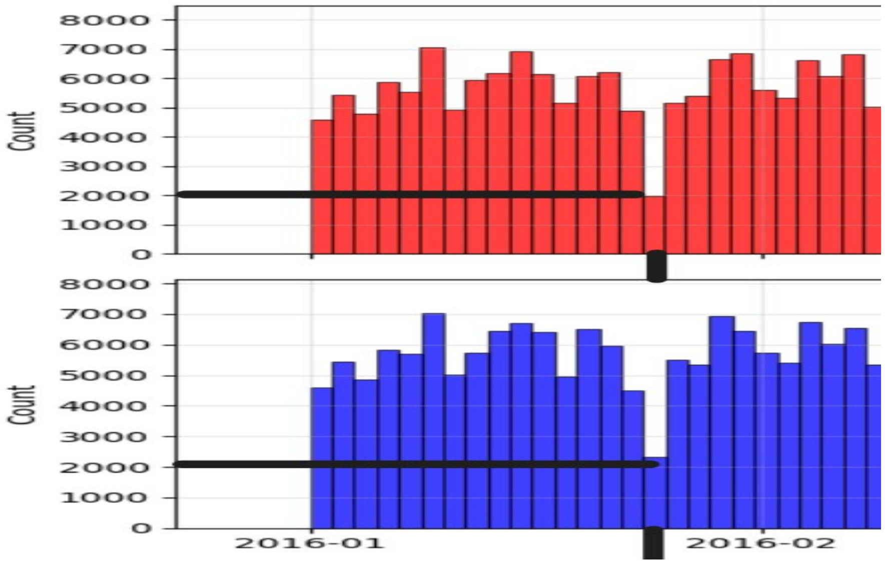


The trip duration distribution is highly skewed to the right, so we visualised it on a log10 scale for the x-axis and a square-root scale for the y-axis. On this scale, most trips look roughly log-normal, with a clear peak just under 1 000 seconds (around 15–20 minutes). At the same time, there are many very short trips under 10 seconds that are almost certainly data errors, and a sharp spike just below 10^5 seconds.

Pickup/Dropoff Distribution over time with count.

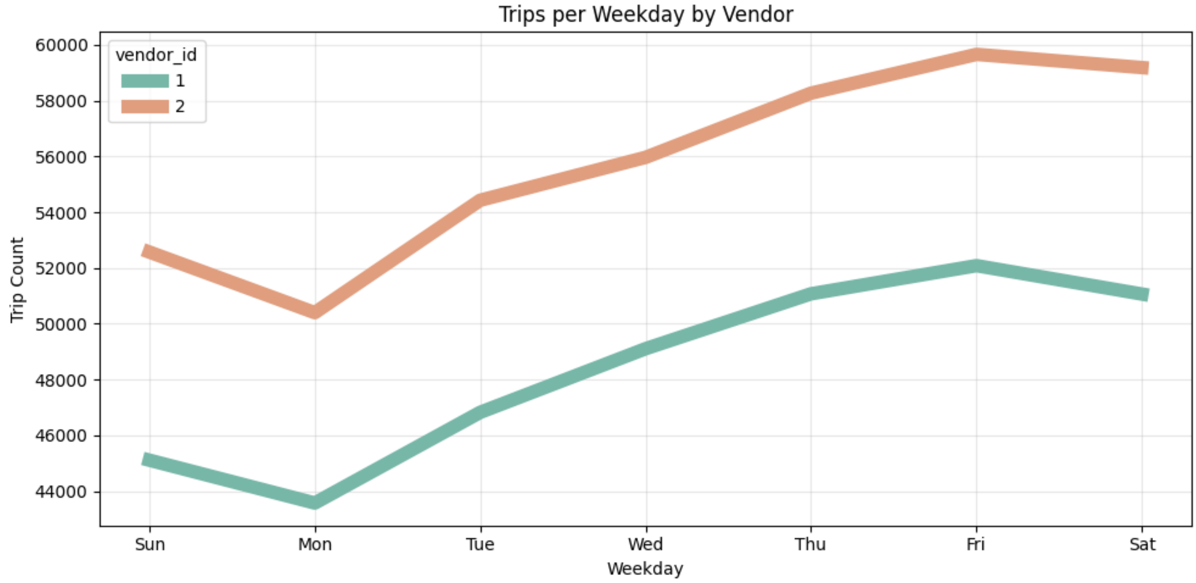


Looking at pickup and dropoff dates, the data spans roughly six months, from January to June. The overall number of trips per day is stable, but not completely flat.

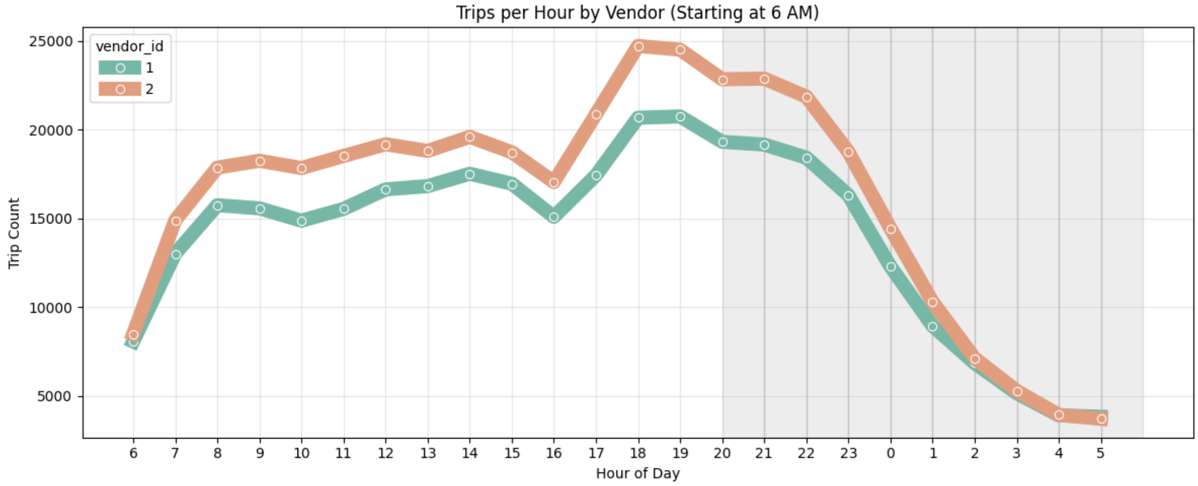


There is a noticeable drop near the end of January, where the daily count falls to around 2000 trips. Without an external reference (e.g. weather or event data) we can only speculate about the cause, but it highlights how real-world factors like storms, holidays, or city events can influence demand.

Trip Per Weekday by Vendor.

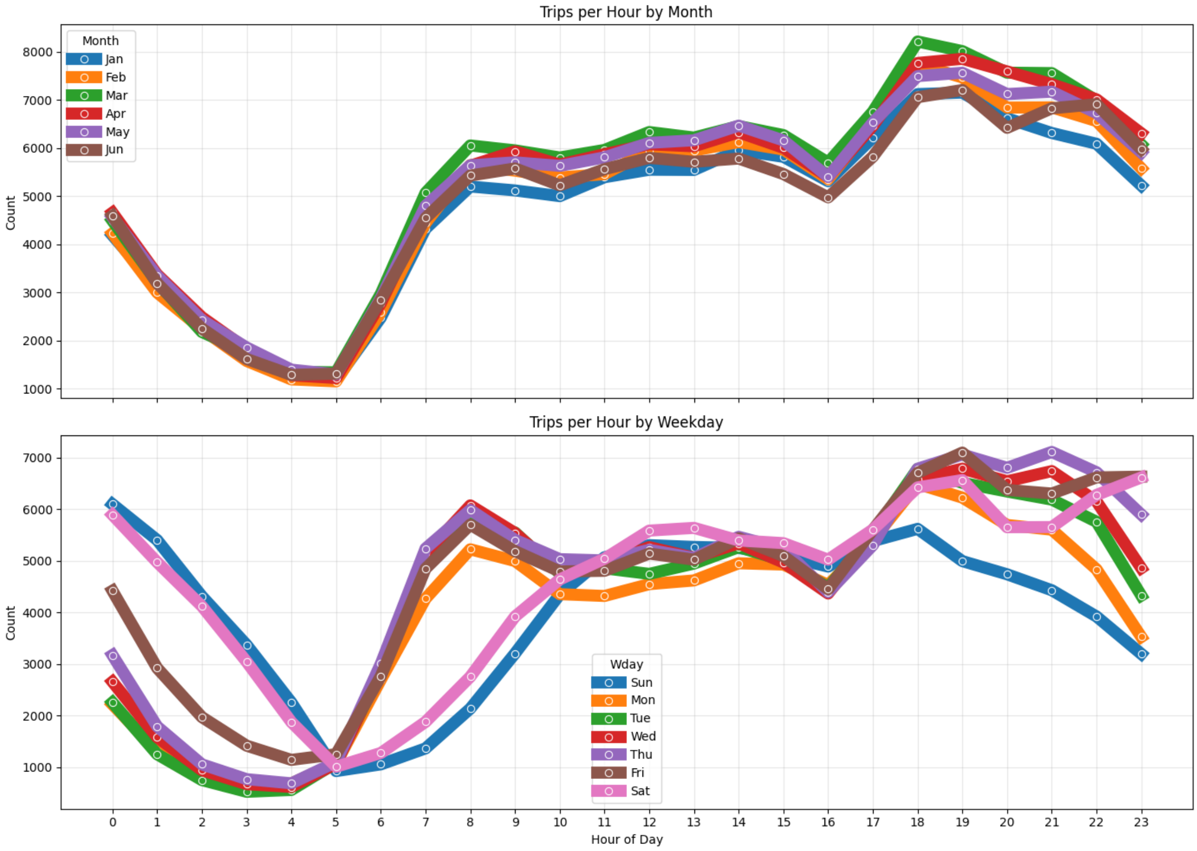


Trip per Hour by Vendor



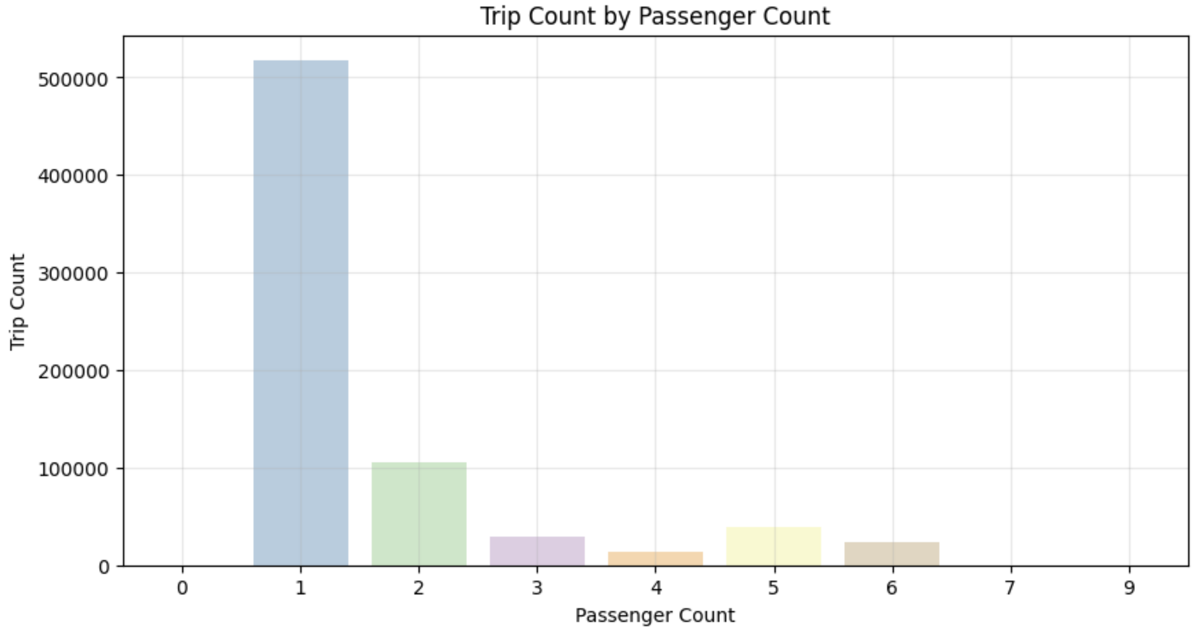
When we break trips down by weekday, both vendors follow almost the same pattern: the lowest activity is on Sunday and Monday, then trip counts gradually rise through the week and peak towards Friday and Saturday. Looking at trips by hour of day, there is a clear daily rhythm. Activity is low in the very early morning (around 4–5 AM), then grows through the morning and afternoon, and reaches its maximum in the evening. Demand then drops sharply after midnight.

Trip per Hour by Month/Weekday



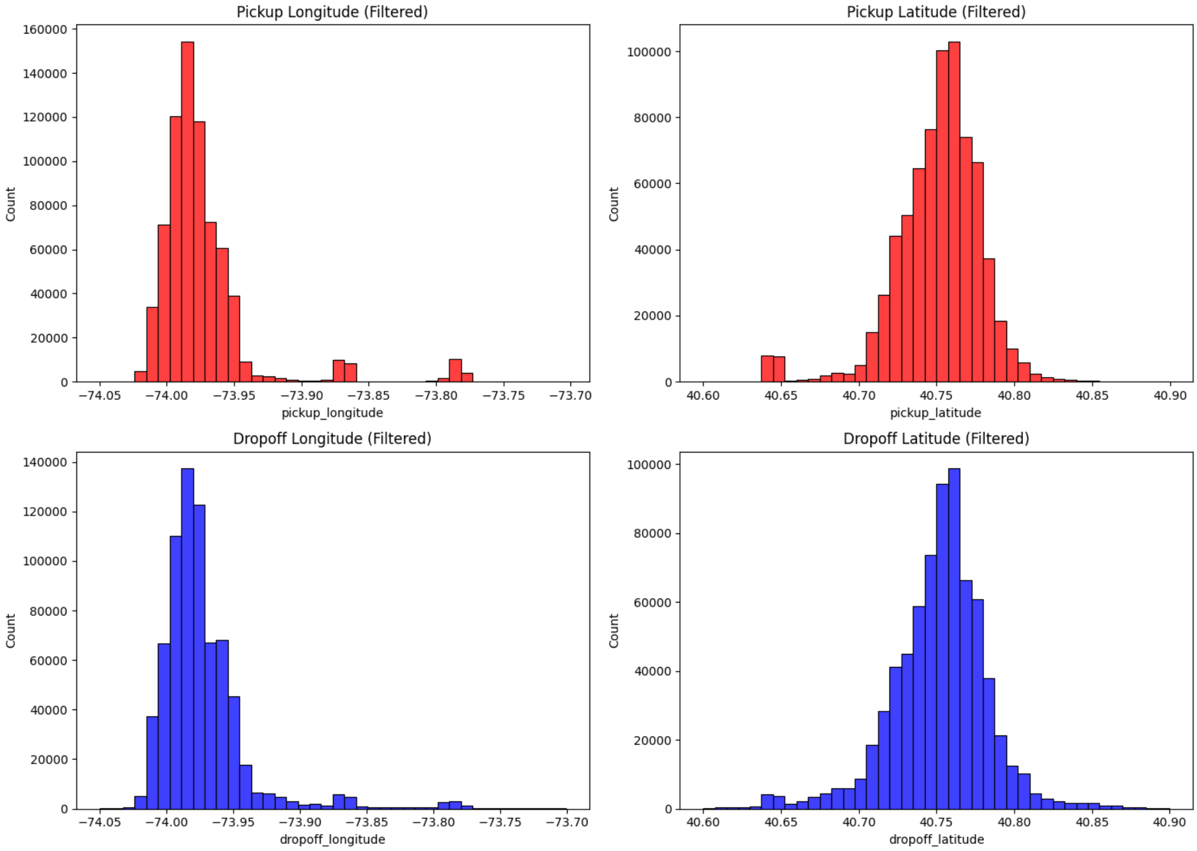
Plotting trips per hour for each month and each day of the week shows that January and June are generally quieter, while March and April are the busiest months. Weekends behave differently from weekdays: late-night and early-morning hours (especially on Friday and Saturday) are much more active because of nightlife, but the classic weekday morning commute peak is weaker on weekends. There is also a visible drop-off in activity on Sunday evening as the weekend ends.

Trip Count by Passenger Count



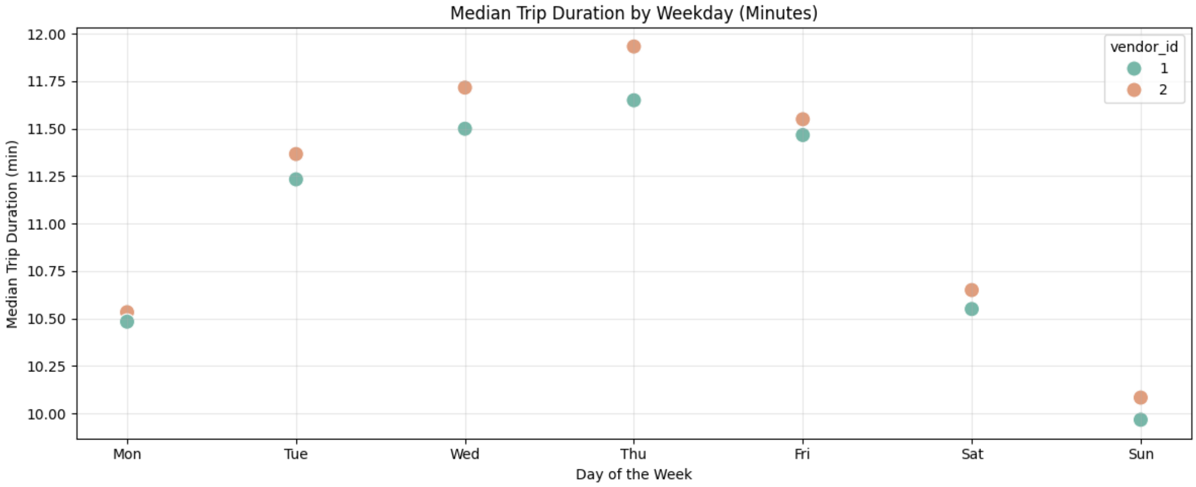
Looking again at passenger count but this time from a frequency point of view, almost all trips have passenger_count between 1 and 3, and only a tiny fraction of trips have 0, 7, or 9 passengers. This supports the decision to treat those rare values as anomalies rather than genuine behaviour.

Spatial Data Distribution Pickup and Dropoff



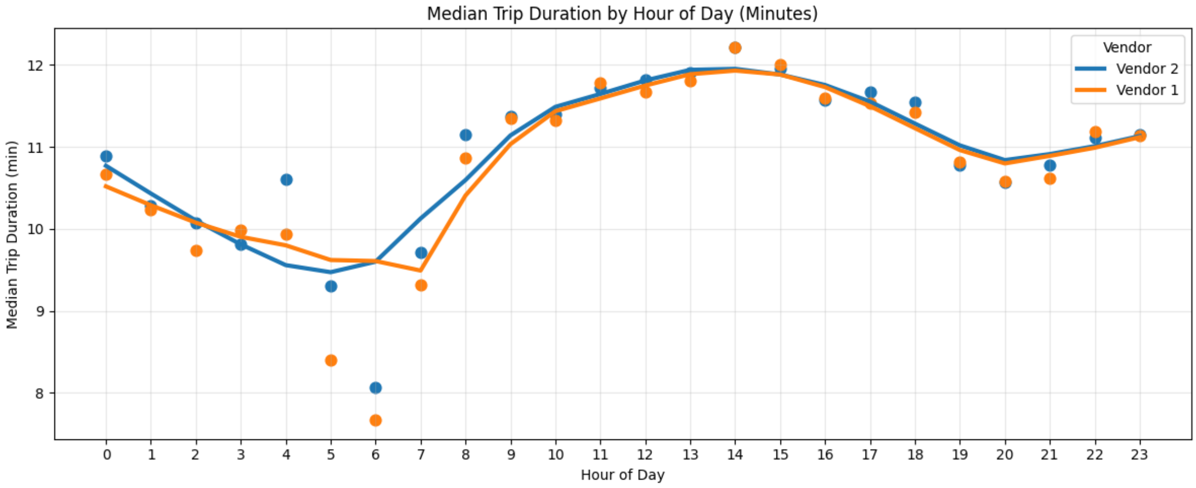
In terms of geography, the pickup and dropoff latitude/longitude plots (after filtering out a handful of points clearly outside New York City) show that most trips are concentrated around Manhattan, with clear clusters near major transport hubs such as JFK and LaGuardia airports. We keep the spatial outliers inside NYC boundaries to avoid biasing the model only towards central Manhattan behaviour.

Median Trip Duration by Weekday (Min)



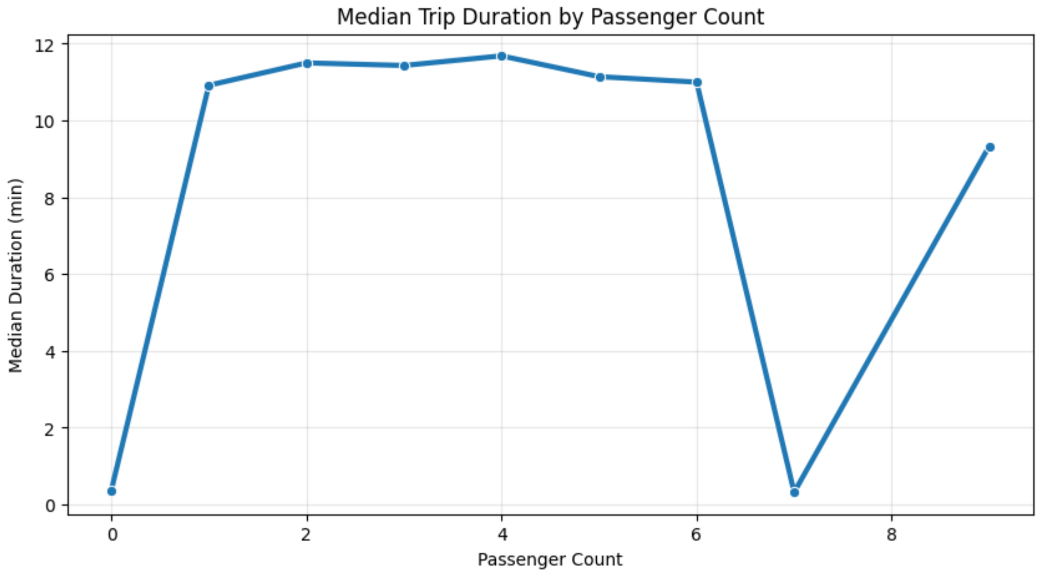
For the bivariate analysis with respect to trip duration, the median duration by weekday shows a smooth pattern: starting around roughly 10.5 minutes on Monday, climbing through the week and peaking on Thursday at about 12 minutes, then dropping to the lowest median on Sunday. Vendor 2 usually has slightly longer median trip durations than Vendor 1, but the difference is small.

Median Trip Duration by Hour of Day (Min)



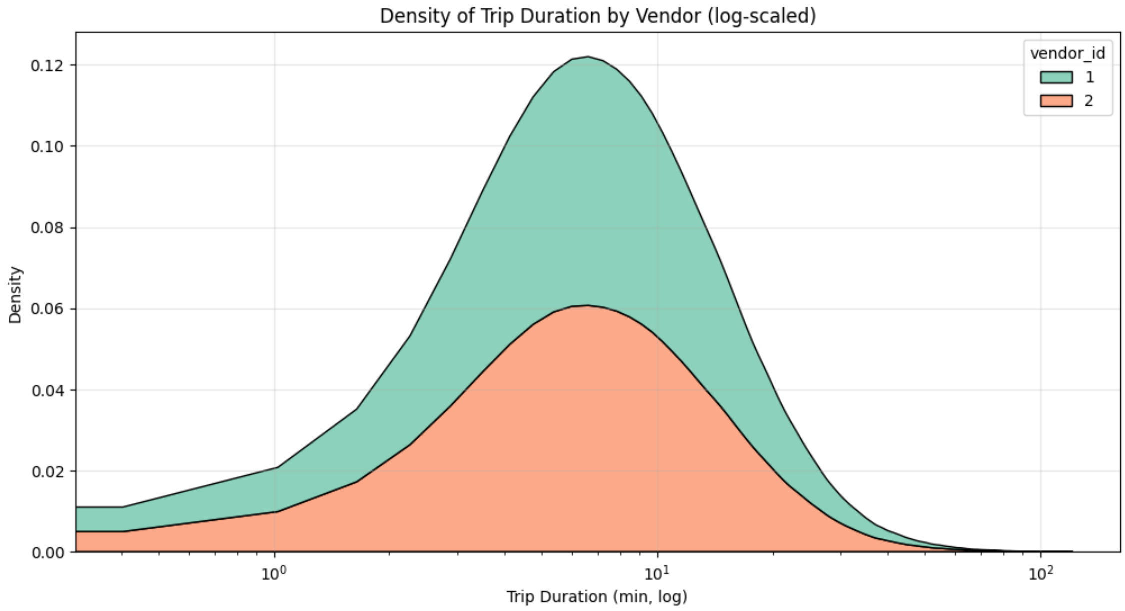
When we look at median duration by hour of day, durations tend to be shortest around 5–7 AM and longer in the early afternoon, which suggests that time-of-day and weekday together are useful features for predicting trip length.

Median Trip Distribution by Passenger Count



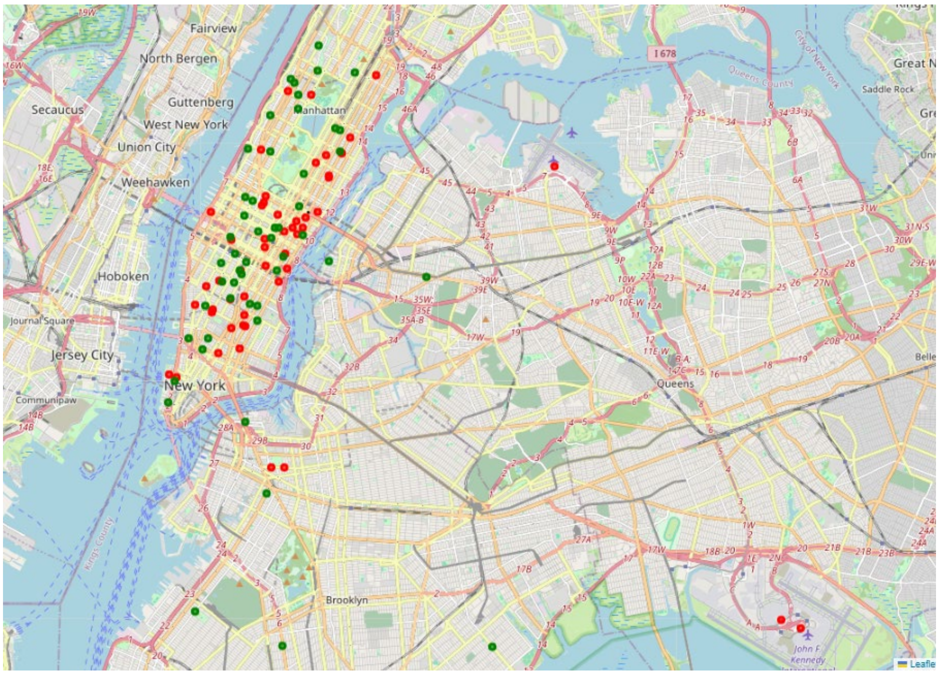
In contrast, when we compare trip duration distributions across passenger counts (ignoring the extreme outliers), the curves are very similar; passenger count doesn't seem to have a strong effect on how long a trip takes.

Density of Trip Duration by Vendor

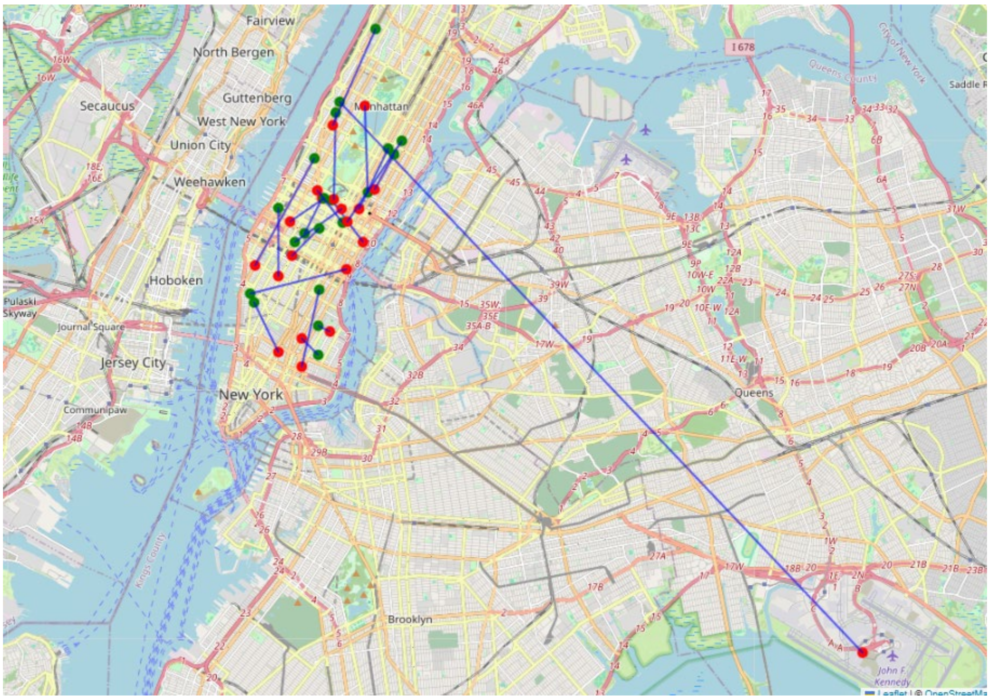


Finally, comparing trip duration for Vendor 1 vs Vendor 2 shows almost no meaningful difference. The two duration density curves almost overlap: both peak in the 5–15-minute range and share the same long tail of longer journeys. This reinforces the idea that vendor_id does not significantly influence trip duration and is more of an administrative field than a behavioural one.

Visual View of Pickup point (Red) and drop off (Green) point.

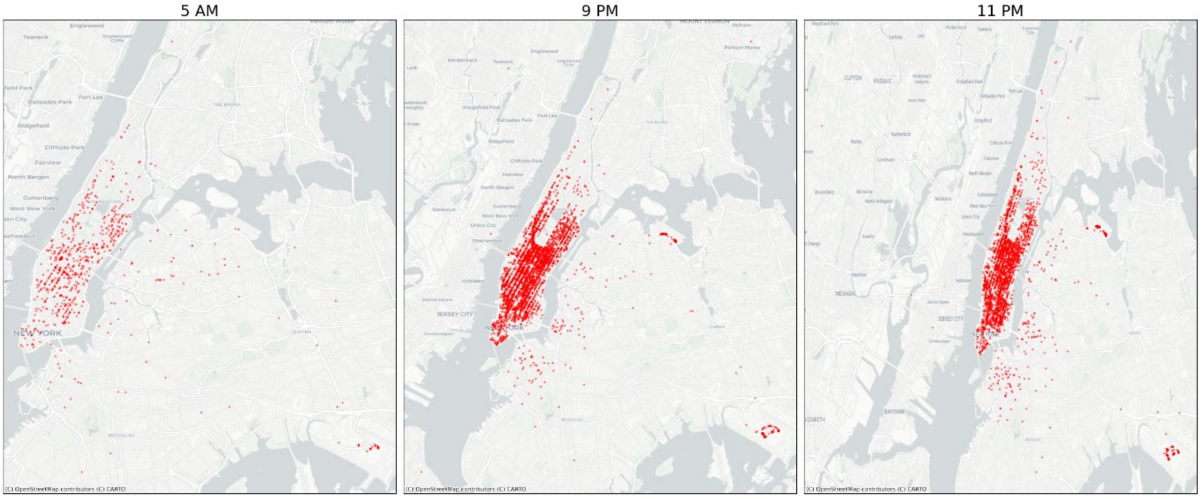


Visual View Connection Pickup and Drop off



Spatial visualisations add another layer to the story. Sampling a subset of pickup and dropoff points and plotting them on a map shows dense activity in Manhattan, with clear connections to the airports. Joining pickup and dropoff locations with straight lines is obviously a simplification real trips follow curved roads, bridges, and tunnels but it still gives a helpful picture of typical origin destination patterns.

Comparison of spatial view across different hour of the day.

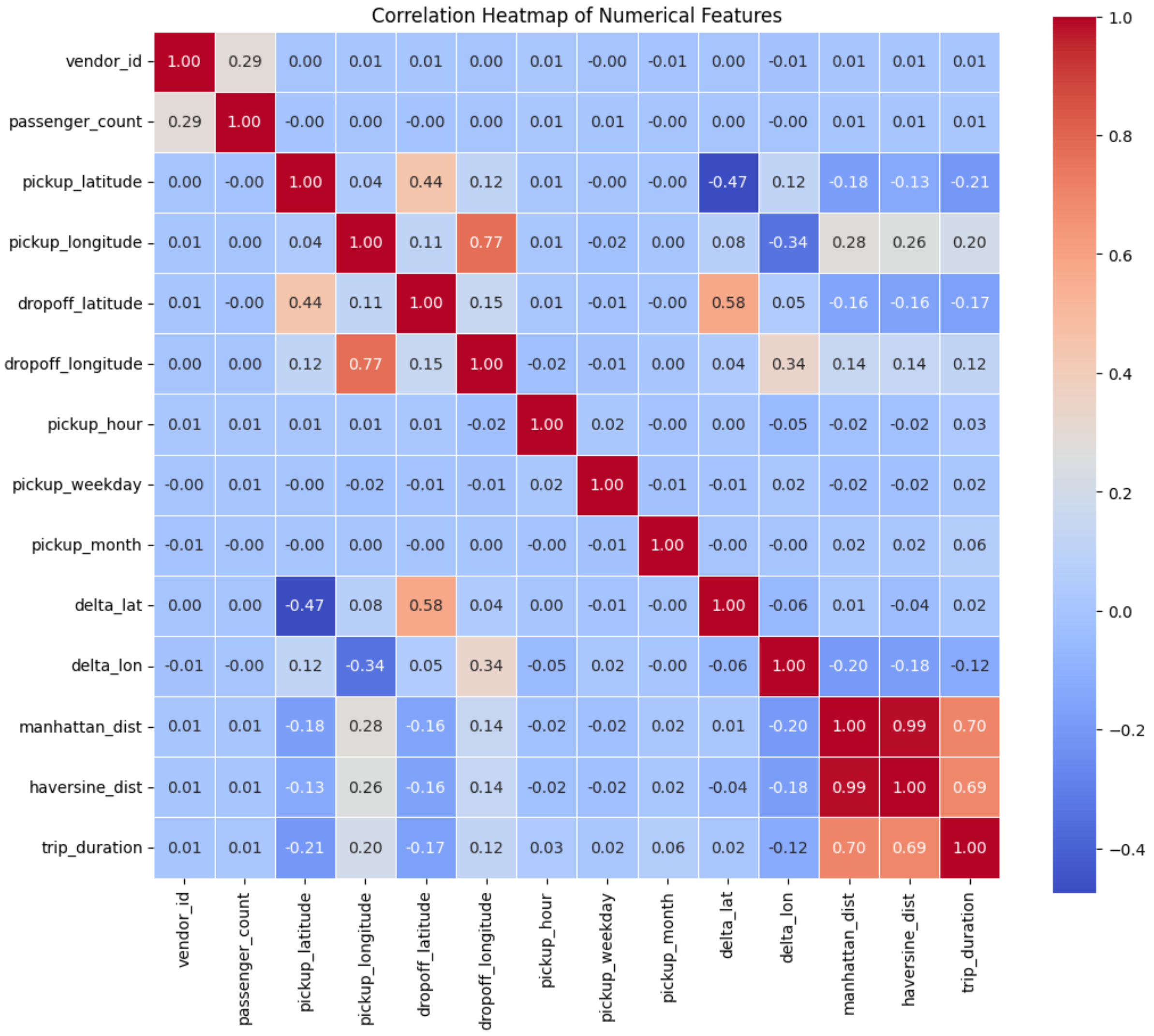

5 AM
9 PM
11 PM
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO

When plot spatial view across different hours of the day (e.g. 5 AM, 9 PM, and 11 PM), we see pickup activity move from sparse, scattered early-morning trips to very dense evening clusters around midtown and lower Manhattan, then gradually tapering off later at night. Even with only a sample of about 10 % of the data, the core spatial structure of demand is very clear.

Overall, the EDA shows that NYC taxi behaviour is strongly driven by distance, time of day, weekday, and location, while factors like vendor ID and passenger count play a much smaller role. These insights directly guide which features are worth engineering and feeding into the predictive models.

Modelling Approach

In this stage we build models to predict trip_duration using the cleaned and engineered features. We begin with a simple Linear Regression model as the baseline, then move to a Random Forest Regressor to capture nonlinear patterns that the linear model cannot learn. All evaluation is done on an unseen 20% test split.

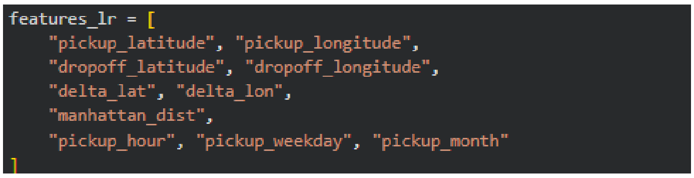


Correlation Heat map

Passenger count and vendor_id showed near-zero correlation with trip duration, so they were dropped. Haversine and Manhattan distances were highly collinear; we kept Manhattan distance because it aligns with NYC's grid layout and correlates more strongly with the target. Delta_lat and delta_lon were kept preserving directional and spatial variation. Pickup and dropoff coordinates show moderate correlation but still contribute useful geographic signal.

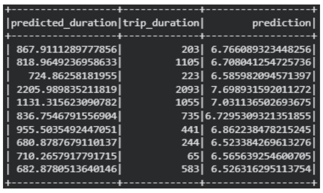
Baseline Linear Regression

Linear Regression is used here mainly as a benchmark. Because trip duration is heavily right-skewed, the model was trained on the log10 of trip_duration instead of raw seconds. The feature vector included spatial and temporal features (coordinates, deltas, Manhattan distance, pickup_hour, weekday, month) but excluded vendor_id and passenger_count because correlation analysis showed they provide almost no predictive value.



```
features_lr = [
    "pickup_latitude", "pickup_longitude",
    "dropoff_latitude", "dropoff_longitude",
    "delta_lat", "delta_lon",
    "manhattan_dist",
    "pickup_hour", "pickup_weekday", "pickup_month"
]
```

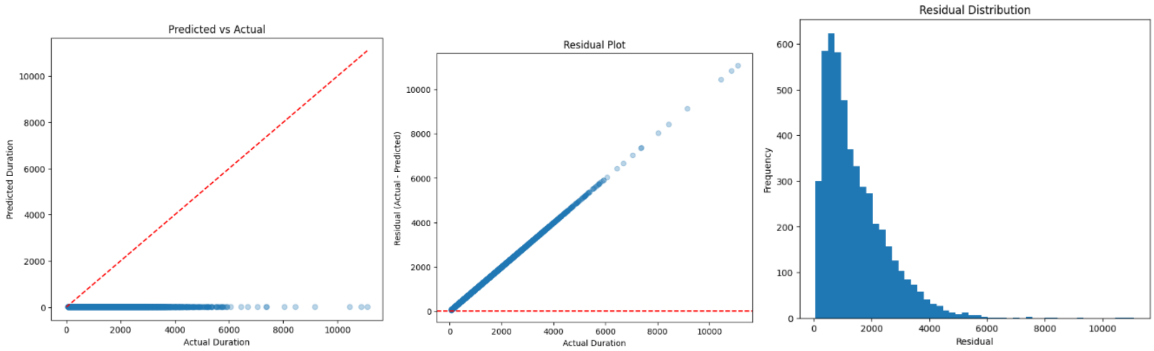
Prediction



```
+------------------+-------------+------------------+
|predicted_duration|trip_duration|        prediction|
+------------------+-------------+------------------+
| 867.9111289777856|          203| 6.766089323448256|
| 818.9649236958633|         1105| 6.708041254725736|
|   724.86258181955|          223| 6.585982094571397|
| 2205.989835211819|         2093| 7.698931592011272|
| 1131.315623090782|         1055| 7.031136502693675|
| 836.7546791556904|          735|6.7295309321351855|
| 955.5035492447051|          441| 6.862238478215245|
| 680.8787679110137|          244| 6.523384269613276|
| 710.2657917791715|           65| 6.565639254600705|
| 682.8780513640146|          583| 6.526316295113754|
+------------------+-------------+------------------+
```

After assembling features and splitting the data (80/20), the model was trained and tested. The baseline achieved:

- MAE (log scale): 0.442
- RMSE (log scale): 0.590056445243076
- R² (log scale): 0.339



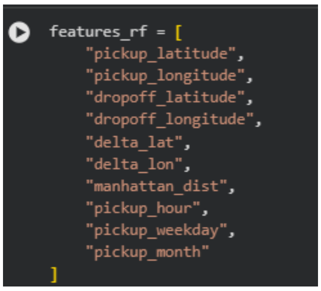
The Linear Regression model, fitted on log-transformed trip duration, demonstrates limited predictive strength. With an MAE of 0.44 (log scale) and an R² of 0.339, it explains only a small portion of the variance, and the diagnostic plots clearly reinforce this weakness. The Predicted-vs-Actual plot shows that the model outputs almost the same small value for all trips, causing predictions to collapse near zero rather than follow the diagonal. This reflects severe underfitting and the model's inability to scale with longer trip durations.

The residual plot displays a near-perfect diagonal pattern, indicating that errors grow in direct proportion to the actual duration. This means the model systematically underpredicts medium and long trips, rather than producing a random spread of errors around zero. The residual distribution is heavily right-skewed, with most small errors clustered near zero but a long tail of very large positive residuals. These correspond to long trips that the model fails to capture, highlighting violations of linearity and homoscedasticity.

Together, these metrics and visual patterns confirm that Linear Regression is unsuitable for modelling NYC taxi trip duration. The underlying relationships are highly nonlinear—driven by spatial patterns, traffic variation, and directional movement—which a simple linear model cannot represent. As a result, it consistently underestimates true durations, particularly for longer and rarer trips.

Random Forest Regression

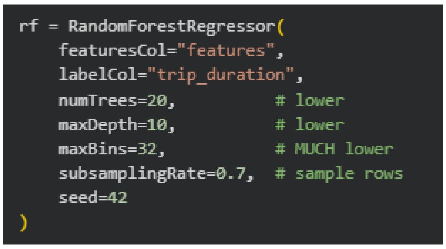
To model nonlinear patterns, we used a Random Forest Regressor with features:



```
features_rf = [
    "pickup_latitude",
    "pickup_longitude",
    "dropoff_latitude",
    "dropoff_longitude",
    "delta_lat",
    "delta_lon",
    "manhattan_dist",
    "pickup_hour",
    "pickup_weekday",
    "pickup_month"
]
```

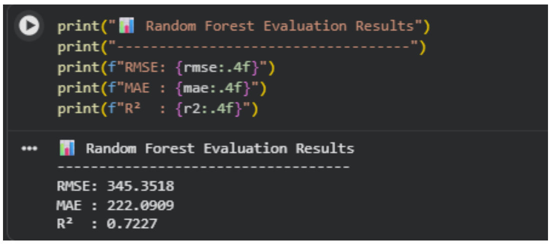
Manhattan distance was used instead of Haversine distance because both are almost perfectly collinear (~0.99 correlation), and Manhattan distance aligns better with NYC's grid-style travel.

Training was done on dataset with an 80/20 train–test split.



```
rf = RandomForestRegressor(
    featuresCol="features",
    labelCol="trip_duration",
    numTrees=20,           # lower
    maxDepth=10,           # lower
    maxBins=32,            # MUCH lower
    subsamplingRate=0.7,   # sample rows
    seed=42
)
```

The Random Forest was configured with lightweight hyperparameters to keep training efficient while still capturing nonlinear patterns. Using 20 trees with maximum depth 10 provides a balanced level of complexity without overfitting. The setting of maxBins=32 speeds up continuous-feature splitting, and a subsamplingRate of 0.7 ensures that each tree trains on a different fraction of the data, improving generalisation and reducing computation. Overall, these settings allow the model to learn meaningful distance- and time-based relationships while remaining computationally manageable on PySpark.



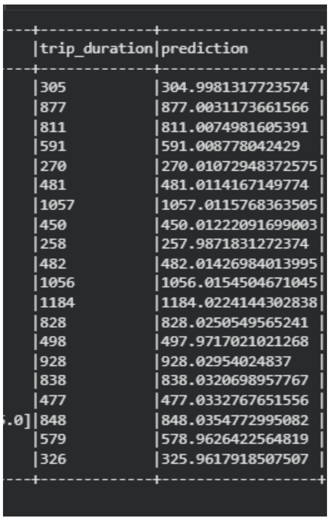
```
print("📊 Random Forest Evaluation Results")
print("----------------------------------")
print(f"RMSE: {rmse:.4f}")
print(f"MAE : {mae:.4f}")
print(f"R²  : {r2:.4f}")
```

```
📊 Random Forest Evaluation Results
----------------------------------
RMSE: 345.3518
MAE : 222.0909
R²  : 0.7227
```

Random Forest Evaluation Results

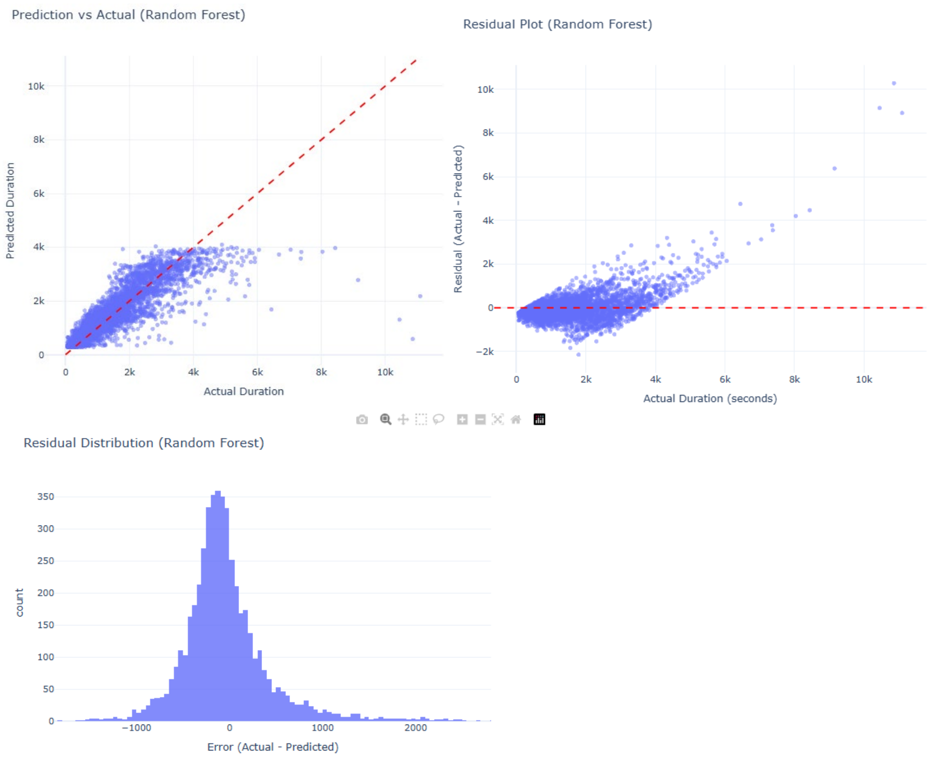
- RMSE: 345.35 seconds
- MAE: 222.09 seconds
- R^2: 0.7227

Prediction



```
---+-------------+-----------------+
   |trip_duration|prediction       |
---+-------------+-----------------+
   |305          |304.9981317723574 |
   |877          |877.0031173661566 |
   |811          |811.0074981605391 |
   |591          |591.008778042429  |
   |270          |270.01072948372575|
   |481          |481.0114167149774 |
   |1057         |1057.0115768363505|
   |450          |450.01222091699003|
   |258          |257.9871831272374 |
   |482          |482.01426984013995|
   |1056         |1056.0154504671045|
   |1184         |1184.0224144302838|
   |828          |828.0250549565241 |
   |498          |497.9717021021268 |
   |928          |928.02954024837   |
   |838          |838.0320698957767 |
   |477          |477.0332767651556 |
.0]|848          |848.0354772995082 |
   |579          |578.9626422564819 |
   |326          |325.9617918507507 |
---+-------------+-----------------+
```

Below are the trips where the model achieves its most accurate predictions.
Note: This is selective prediction to see the best results.



The Random Forest model shows strong predictive ability and clearly adapts to the nonlinear nature of NYC travel dynamics. The Predicted-vs-Actual plot forms a tight diagonal cloud, indicating that the model scales its predictions well for short and medium trips, unlike Linear Regression which collapsed near zero. Some underestimation appears for very long trips, but this is expected because such cases are rare, and tree ensembles naturally pull extreme values toward the mean.

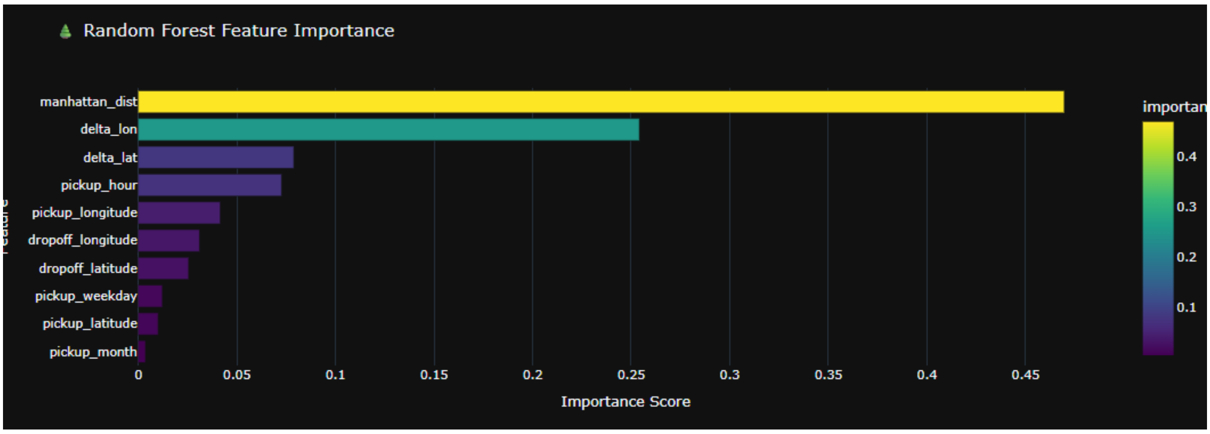
The residual plot confirms stable performance: errors remain centred around zero with no systematic pattern, and both over- and under-predictions are present in reasonable balance. Only long-duration outliers produce larger residuals. The residual distribution is roughly symmetric and bell-shaped, showing that most errors are small and that the model does not suffer from the skewed, expanding error behaviour observed in Linear Regression.

Results and Discussion

Linear Regression	Random Forest Regression
R^2 (log scale): 0.339	R^2: 0.7227
RMSE (log scale): 0.590056445243076	RMSE: 345.3518
MAE (log scale): 0.442	MAE: 222.09 seconds

The side-by-side comparison between the models clearly shows that Random Forest substantially outperforms Linear Regression for this prediction task. While the linear model fails to scale with increasing trip duration and systematically underestimates longer trips, the Random Forest captures the underlying nonlinear relationships and produces predictions that align much more closely with the true values. Its residuals are centred, stable, and far less biased, whereas Linear Regression displays expanding errors and strong skewness. The Random Forest's tighter diagonal pattern in the Predicted-vs-Actual plot, along with its symmetric residual distribution, demonstrates far superior generalisation and makes it the more appropriate choice for modelling the complex spatial–temporal dynamics of NYC trip durations.

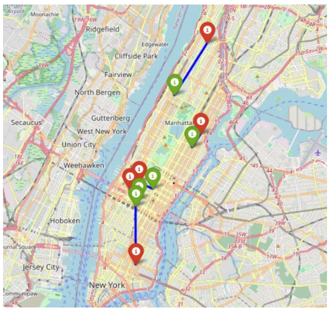
Interpretation of Feature Importance

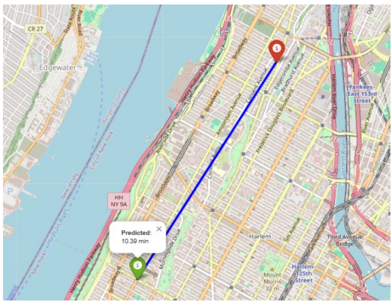


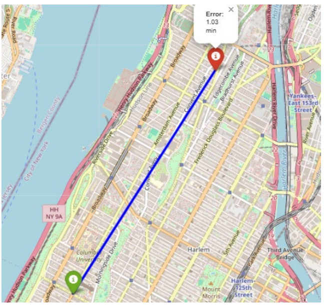
The feature-importance results show that *manhattan_dist* is by far the strongest predictor of trip duration, reflecting how grid-based travel distance directly influences travel time in NYC. The spatial deltas (*delta_lon* and *delta_lat*) also rank highly, indicating that directional movement and geographic separation add meaningful predictive signal. Temporal features such as pickup hour contribute moderately, capturing daily traffic fluctuations. Coordinate values and categorical time indicators (weekday, month) play smaller roles, suggesting they refine predictions but do not dominate them. Overall, the model relies primarily on spatial geometry, with temporal and location-specific variables providing secondary adjustments.

High-Level Prediction Demonstration

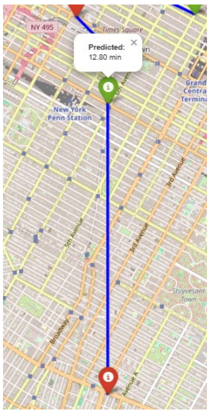
To illustrate how the model behaves on real-world trips, we selected five examples from the unseen test set and visualised them on a map. For each trip, the pickup location displays the model's predicted duration, while the dropoff location shows the actual error (difference between predicted and true trip time). This provides an intuitive understanding of how well the model generalises to new data.

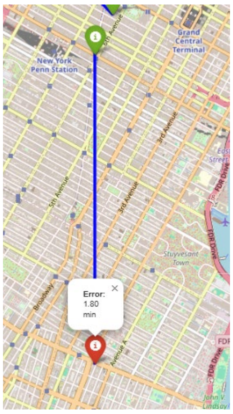






Most predictions were within 1–3 minutes of the actual duration, which aligns with the overall performance metrics of the Random Forest model. Larger errors tended to occur on longer or more irregular trips, which is expected since real travel times are influenced by traffic, driver behaviour, and routing choices — factors not fully captured in the dataset.





The visualisation helps non-technical readers grasp what the model is doing: it estimates how long a ride might take, highlights where errors occur and shows how closely the estimated route corresponds to the actual pickup and dropoff points. This makes the predictive model easier to interpret and demonstrates its practical usefulness in providing reasonable time estimates for taxi trips.

Limitations and Future Work

While the model performs reasonably well, several limitations remain due to the scope of the dataset. The features available only describe the trip itself (pickup time, location, and rough distance), but do not include the external factors that heavily influence real-world travel time. Important signals such as weather conditions, traffic congestion, road closures, taxi speed patterns, and special events are missing, and these can easily shift trip duration by several minutes. Because of this, even the Random Forest model cannot fully explain the variability in longer or irregular trips.

Another limitation is that the dataset covers only a six-month period, which may not capture seasonal changes beyond that window. The coordinate system also has occasional GPS errors, and the model treats the city as a simple grid, ignoring actual road networks, traffic directions, and route choices. Using Manhattan distance helps, but it is still an approximation rather than a true travel-path estimate.

Conclusion

This project explored the NYC Taxi Trip Duration dataset from initial examination through cleaning, feature engineering, exploratory analysis, and predictive modelling. We identified and removed unrealistic duration values, corrected rare passenger-count anomalies, and engineered meaningful features such as Manhattan distance and time-based variables. EDA revealed clear patterns in demand across hours, weekdays, and locations, and showed that distance and time-of-day are the strongest drivers of duration.

Using these insights, we built two models: a baseline Linear Regression and a more flexible Random Forest Regressor. The Random Forest clearly outperformed the linear model, achieving an R^2 of about 0.72 on unseen data and reducing prediction errors noticeably. Feature importance confirmed that Manhattan distance, pickup hour, and weekday contribute most to the model’s accuracy.

Overall, the study demonstrates that even with a limited set of features, solid preprocessing, spatial/temporal engineering, and a non-linear model can generate useful duration estimates for NYC taxi trips. With additional external data and more advanced models, there is strong potential to push the performance further and produce more reliable, real-world-ready predictions.