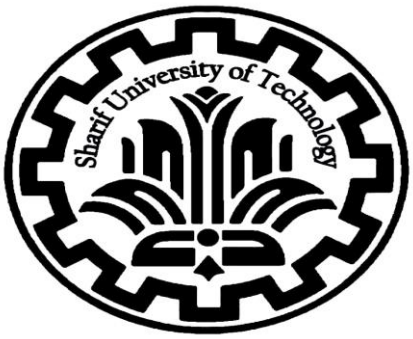

Sharif University of Technology

Phase Two of the Transportation Planning Course Project

Authors:

Amirhossein Pourkarimi

Mahshad Ebrahimi

Ali Bozorgmehry

Course Instructor:

Professor Hasan Naeibi

Winter 2025

Data Preprocessing:

To ensure precise and clean data preprocessing, we conducted this phase in a separate Jupyter Notebook file from the main file. The file Phase_Two_Preprocessing contains the operations for this phase. In this file, after identifying the raw input files, unnecessary columns were removed from them. Additionally, we removed the records that appeared to be mistakenly registered (for example, rows where the recorded traveled distance was less than or equal to zero).

Data that needed to be extracted based on timestamps were also processed. (It's worth noting that these extracted data might be needed in later stages, and we stored date and time columns in another format within the DataFrame. This is because handling missing values—NaN—using the Random Forest algorithm requires properly formatted timestamps. Therefore, we created copies of these date-time columns with appropriate formatting to ensure their usability later.) The remainder of the data preprocessing steps aligns with those from the previous phase.

Furthermore, in this preprocessing step, the time frames are grouped into categories. Since, on average, in the previous phase, three specific time periods were identified during the day (as evident in the diagram below), we grouped the time frames into three categories again: low traffic, moderate traffic, and high traffic.

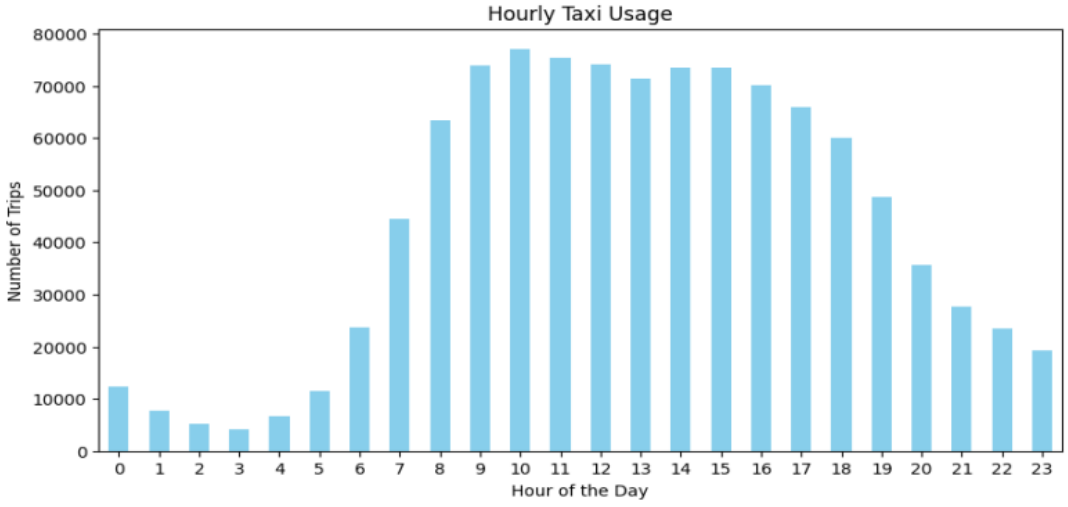

Figure 1 Graph of Trip Counts During Different Hours of the Day

After completing the preprocessing steps, the result is saved in a file named Preprocessed.csv so that any required modifications in the future can be easily managed. Subsequently, the tasks are continued in the file Phase_Two.

First Requirement:

1. In this file, we first import the Preprocessed.csv file and then proceed to analyze the requirements. For different city regions (Boroughs), an OD matrix must be calculated. To do this, we also import the Boroughs.csv file and add the names of the boroughs to the corresponding DataFrame. Afterward, we calculate the OD matrix for trips between the different boroughs.

Borough OD matrix:

DO_Borough / PU_Borough	Bronx	Brooklyn	EWR	Manhattan	Queens	Staten Island
Bronx	26110.0	2941.0	4.0	11117.0	3254.0	275.0
Brooklyn	2884.0	74754.0	140.0	14638.0	8829.0	877.0
EWR	0.0	0.0	2.0	3.0	0.0	0.0
Manhattan	21297.0	5478.0	110.0	272701.0	8711.0	228.0
Queens	3492.0	8728.0	11.0	12373.0	94121.0	227.0
Staten Island	239.0	615.0	4.0	225.0	236.0	213.0

2. Afterward, we determine which two regions (with distinct origins and destinations) have the highest number of trips. The results are displayed in the code under the label External Trips. The trips with the highest frequency occur between Manhattan to Bronx, Manhattan to Queens, and Manhattan to Brooklyn in descending order.

```
External Trips
Manhattan to Bronx: 19.92%
Brooklyn to Manhattan: 13.69%
Queens to Manhattan: 11.57%
Bronx to Manhattan: 10.40%
Brooklyn to Queens: 8.26%
Queens to Brooklyn: 8.16%
Manhattan to Queens: 8.15%
Manhattan to Brooklyn: 5.12%
Queens to Bronx: 3.27%
Bronx to Queens: 3.04%
Bronx to Brooklyn: 2.75%
Brooklyn to Bronx: 2.70%
Brooklyn to Staten Island: 0.82%
Staten Island to Brooklyn: 0.58%
Bronx to Staten Island: 0.26%
Staten Island to Bronx: 0.22%
Staten Island to Queens: 0.22%
Manhattan to Staten Island: 0.21%
Queens to Staten Island: 0.21%
Staten Island to Manhattan: 0.21%
Brooklyn to EWR: 0.13%
Manhattan to EWR: 0.10%
Queens to EWR: 0.01%
Bronx to EWR: 0.00%
Staten Island to EWR: 0.00%
EWR to Manhattan: 0.00%
EWR to Bronx: 0.00%
EWR to Staten Island: 0.00%
```

3. Next, we perform the same analysis for intra-regional trips (where origin and destination are the same), labeled as Internal Trips. The regions with the highest number of trips, in descending order, are Manhattan, Queens, and Brooklyn.
4. Considering that a significant percentage of trips occur in Manhattan, demand in this area is high. Additionally, Brooklyn and Queens are also attractive points for investment. However, due to the large number of trips in these regions, competition is certainly tough. Nonetheless, the high demand creates opportunities for us to grow our business effectively.

Second Requirement:

Given that a fixed pricing strategy of has been established, only routes with an average cost greater than or equal to $40 should be considered. (This is because if a route's cost is below $40, offering services for it at $40 would mean covering a route worth less than $40 for $40, which could result in customers refraining from using our services for such routes). Additionally, it is essential that the average cost of a route is greater than or equal to $40. Simply removing all rows with costs less than $40 from the dataset could lead to mistakes. This is because a route could, in reality, have an average cost of $35 (but, in two rows of data, $50 and $55 were paid for those trips, while all remaining rows for this route are under $35). Removing all rows below $35 would cause the route's average cost to calculate to $52.5, falsely suggesting it qualifies.

As a result, to avoid such mistakes, we calculate the average cost for each route per row and remove any rows where the route's average cost is less than $40.

Now, we calculate the average daily trips for these regions. Given that the data is for the year 2021, grouping the data by day and month (GroupBy) allows us to compute the daily averages. The OD matrix for all boroughs is calculated. Considering the large dimensions of this matrix, a sub-matrix is identified which, in our view, carries the greatest weight. This sub-matrix calculates the number of trips between the 25 busiest boroughs and is named top_zone_od_matrix.

Then, using the supply-and-demand method, the created matrix is balanced and referred to as balanced_od_matrix.

Following this, for utilizing impedance functions when calling the Gravity Model function, we define exponential and lognormal impedance functions such that their input is a cost matrix, and they return a Deterrence Matrix. Additionally, the Gravity Model executes its algorithm using inputs such as the Cost Matrix and the Deterrence Matrix.

We now identify the indices of the 25 top zones so that the cost matrix for those zones can be calculated using them. It is noteworthy that to compute the cost matrix, we have utilized the weighted average, where the weight of each trip is its distance.

Then, using the Deterrence Matrix and the cost matrix, we execute the Gravity Model. According to the results obtained, the exponential function demonstrates greater accuracy, and for the continuation of the project, we use the matrix derived from the exponential impedance function. The output of this impedance function, namely the OD matrix resulting from the execution of the Gravity Model, is included in an attached file named Main_OD.

Next, we aim to calculate the cost matrix for different time groups (low traffic, normal traffic, and heavy traffic). To calculate each of these matrices, we first filter the data based on the specific time group and then compute the weighted average (where the weight of the cost is again the distance traveled during the trip). As a result, we have three cost matrices categorized based on the different time groups, which are attached as CSV files under the names {Cost_Matrix_Group}.

Additionally, to calculate the distance matrix for all trips between specific origin and destination points, the average distance traveled is calculated. This matrix is included in the file distance_matrix.csv.

Now, using all the above data, we proceed to define the optimization problem.

Optimization

The code for this section is included in the file Optimization. For this problem, the focus is on making predictions in a way that ensures the maximum possible number of vehicles are stationed in profit-generating zones. Hence, the random variable of interest here is Xijt, which represents the number of trips (essentially, the number of vehicles).

To incorporate the influence of tips into the modeling process, the Random Forest model is trained again, since tip prediction (whether a tip is paid or not) was modeled in a prior phase using different features. Additionally, the model's output for determining whether trips made for each route at different time groupings lead to tipping has been saved in the file Important_Locations_Tip_Predictions.csv.

The model's accuracy is then treated as the probability of tipping. The output of the model for each 1 or 0 ijt indicates whether a tip is paid or not:

- 1 implies a tip was paid.
- 0 implies no tip was paid.

In the mathematical modeling process, this outcome is represented as Zijt. The mathematical modeling formulation is described as follows:

Additionally, the optimization code has been implemented in the mentioned file, Optimization.

X_{ijt} → تعداد سفرهای انجام شده از نقطه‌ی i به نقطه‌ی j در بازه زمانی t / که یکی از سه حالت {Low, Normal, Peak} است.

Z_{ijt} → پارامتر صفر و یک که خروجی مدل رندم فارست است

O_{ij} → ماتریس تقاضای سفر از i به j

C_{ijt} → ماتریس هزینه جابه‌جایی از i به j در زمان t

D_{ij} → ماتریس فاصله بین i و j

P → ۴۰ عدد ثابت $\rho =$ احتمال پرداخت انعام

L → ۱۰۰۰ عدد ثابت

$$\text{Max } Z : \sum_t \sum_j \sum_i \left[X_{ijt} \cdot (\rho - C_{ijt}) \right] + \Delta\rho \sum_t \sum_j \sum_i (X_{ijt} \cdot Z_{ijt})$$

s.t.

$$\sum_t X_{ijt} \le O_{ij} \quad \forall i,j$$

این محدودیت بیان می‌کند که تعداد سفرها باید از تعداد تقاضای آن i به j کمتر باشد.

$$\sum_t \sum_i \sum_j (X_{ijt} \cdot D_{ij}) \le L$$

حداکثر مسافت طی شده

$$X_{ijt} \ge 0 \quad \forall i,j,t$$