



**Industrial Engineering
Department
Sharif University of
Technology**

Project Phase 2

Transportation Planning
DR. Hassannayebi

**Amirali Modir 400103635
Amirhossein Monji 400103679**

پیش پردازش داده‌ها:

ابتدا کتابخانه‌های مورد نظر را فراخوانی می‌کنیم. سپس اولین قدم Load کردن داده‌هاست. پس از این کار، سوتون‌هایی که به آن‌ها نیازی نداریم را حذف می‌کنیم. سپس برای هر ستون تعداد nullها را بررسی می‌کنیم.

```
VendorID      249115
lpep_pickup_datetime  0
lpep_dropoff_datetime  0
PULocationID   0
DOLocationID   0
passenger_count 412434
trip_distance   0
tip_amount      0
tolls_amount    0
total_amount    0
payment_type    412434
trip_type       412434
dtype: int64
```

عکس 1 - تعداد خانه‌های خالی در هر ستون

حالا به سراغ بررسی ستون payment_type می‌رویم. مشاهده می‌کنیم که در این ستون 298760 داده از پرداخت نوع یک و تنها یک عدد از پرداخت نوع سوم استفاده کرده‌اند. پس به احتمال زیاد نوع یک همان پرداخت با کارت اعتباری است. پس در نتیجه تمام کسانی که انعام داده‌اند ($tip > 0$)، پرداخت نوع اول را دارند.

```
payment_type
1.0      298760
3.0         1
dtype: int64
```

عکس 2 - انواع پرداخت، هنگامی که انعام پرداخت شده باشد.

حال تعداد مسافره‌ای هر سفر را بررسی می‌کنیم. اگر این عدد بیشتر از 4 بود، ماشین حتماً ون، نوع سفر از نوع Dispatch و نحوه پرداخت نامشخص بوده. پس بعد از فیلتر کردن بر اساس تعداد مسافر، نوع

سفر را مشاهده می‌کنیم. 22695 داده از نوع سفر 1 و 382 عدد از نوع 2 بودند. پس نوع سوم یعنی Dispatch را اضافه می‌کنیم.

```
trip_type
1.0      22695
2.0        382
dtype: int64
```

عکس 3 - انواع سفر، هنگامی که تعداد مسافران بیشتر 4 باشد.

حالا به سراغ بررسی نحوه پرداخت در کل داده‌ها می‌رویم. مشاهده می‌کنیم که از نوع 1 تا 5 بوده. چون نوع پنجم تنها 5 عدد بوده، آن را نامشخص فرض می‌کنیم. پس آن دسته از سفرهایی که تعداد مسافرشان از 4 بیشتر بوده، نحوه پرداخت پنج و نوع سفر سه داشتند.

```
payment_type
1.0      461268
2.0      259274
3.0         3891
4.0         1091
5.0           7
dtype: int64
```

عکس 4 - انواع پرداخت در کل داده‌ها

قبل از اینکه سطرهایی که Vendor ID ندارند را حذف کنیم، تعداد کل داده‌ها را برای محاسبه فاکتور بالانس ذخیره می‌کنیم. حال داده‌هایی که Vendor ID ندارند را از Dataset حذف می‌کنیم. سپس چون تنها داده‌های سال 2021 را نیاز داریم، روی این سال باید فیلتر کنیم. اما قبل از این کار باید Type مربوط به pickup_datetime و dropoff_datetime را تصحیح کرده به Datetime تغییر دهیم و سپس فیلتر را انجام دهیم.

بعد از انجام تمام این کار ها، همچنان Dataset در بعضی نقاط Missing دارد. پس با استفاده از RandomForest ابتدا Feature های اثرگذار بر آن ها را انتخاب کرده (4 Feature بیشترین اثرگذار) و داده های خالی را Predict می کنیم. پس از انجام این کار، خواسته اول سوال تمام شده و داده ما پاکسازی شده. حالا به سراغ خواسته دوم می رویم. ستون های با مقدار خالی passenger_count و payment_type و trip_type هستند. ابتدا با passenger_count شروع کرده و به ترتیب پیش می رویم. در انتخاب feature ها و پیشبینی هر ستون، ستونی که مقادیر خالی دارند را در نظر نمی گیریم.

```
VendorID      0.000000
PULocationID  0.000000
DOLocationID  0.000000
passenger_count 0.199264
trip_distance 0.000000
tip_amount    0.000000
tolls_amount  0.000000
total_amount  0.000000
payment_type   0.163446
trip_type      0.199264
pickup_datetime 0.000000
dropoff_datetime 0.000000
duration       0.000000
dtype: float64
```

عکس 5 - نسبت missing در هر ستون قبل از پیشبینی

در نهایت با استفاده از RandomForest ابتدا Feature های مهم تر برای پر کردن خانه های خالی را انتخاب می کنیم و سپس با استفاده از آن ها و الگوریتم ذکر شده مقادیر خالی را پیشبینی می کنیم. سپس مشاهده می کنیم که کل جدول کامل شده است.

```

VendorID      0
PULocationID  0
DOLocationID  0
passenger_count  0
trip_distance  0
tip_amount     0
tolls_amount   0
total_amount   0
payment_type   0
trip_type      0
pickup_datetime  0
dropoff_datetime 0
duration       0
dtype: int64

```

عکس 6 - نسبت missing در هر ستون بعد از پیشبینی

خواسته اول:

ابتدا داده مربوط به مناطق را Load می‌کنیم؛ سپس اطلاعات در این جدول را با جدول اصلی ادغام می‌کنیم. سپس ماتریس OD برای مناطق را می‌سازیم.

Destination	Bronx	Brooklyn	EWB	Manhattan	Queens	Staten Island
Origin						
Bronx	40329	4926	6	17613	4637	408
Brooklyn	4820	116100	227	24421	14280	1282
EWB	0	0	11	7	0	0
Manhattan	30497	8041	180	360777	12236	336
Queens	4820	13619	21	19639	137736	374
Staten Island	342	869	5	316	381	354

عکس 7 - ماتریس OD مناطق

ماتریس OD بالا بالانس نیست. با استفاده از فاکتور بالانس که پیش تر حساب کردیم، ماتریس را بالانس می کنیم.

'''	Destination	Bronx	Brooklyn	EWB	Manhattan	Queens	Staten Island
	Origin						
	Bronx	52586.730915	6423.224887	7.823660	22966.354028	6046.385263	532.008882
	Brooklyn	6285.006894	151387.821647	295.995138	31843.600279	18620.310880	1671.655360
	EWB	0.000000	0.000000	14.343377	9.127603	0.000000	0.000000
	Manhattan	39766.360007	10485.008388	234.709801	470432.765980	15955.050695	438.124962
	Queens	6285.006894	17758.404333	27.382810	25608.143233	179599.939727	487.674809
	Staten Island	445.948622	1133.126762	6.519717	412.046095	496.802412	461.595942

عکس 8 - ماتریس OD مناطق بالانس شده

سپس درصد سفر از هر منطقه به هر منطقه را محاسبه می کنیم.

	Destination	Bronx	Brooklyn	EWB	Manhattan	Queens	Staten Island
	Origin						
	Bronx	4.920511	0.601018	0.000732	2.148949	0.565757	0.049780
	Brooklyn	0.588085	14.165274	0.027696	2.979588	1.742292	0.156416
	EWB	0.000000	0.000000	0.001342	0.000854	0.000000	0.000000
	Manhattan	3.720916	0.981076	0.021962	44.018131	1.492905	0.040995
	Queens	0.588085	1.661644	0.002562	2.396140	16.805066	0.045631
	Staten Island	0.041727	0.106026	0.000610	0.038555	0.046486	0.043191

عکس 9 - ماتریس درصد سفر بین مناطق

سپس با مرتب کردن درصدهای بالا بیشترین سفر میان دو بخش متفاوت و یکسان را پیدا می‌کنیم.

```
... Origin: Manhattan , Destination: Bronx , Percentage 3.720916045436244
Origin: Brooklyn , Destination: Manhattan , Percentage 2.979587852759239
Origin: Queens , Destination: Manhattan , Percentage 2.3961396273837554
Origin: Bronx , Destination: Manhattan , Percentage 2.1489488903258867
Origin: Brooklyn , Destination: Queens , Percentage 1.7422920657385832
Origin: Queens , Destination: Brooklyn , Percentage 1.6616439526116078
Origin: Manhattan , Destination: Queens , Percentage 1.4929051622112954
Origin: Manhattan , Destination: Brooklyn , Percentage 0.9810763655885115
Origin: Bronx , Destination: Brooklyn , Percentage 0.6010175571308305
Origin: Queens , Destination: Bronx , Percentage 0.5880845768109223
Origin: Brooklyn , Destination: Bronx , Percentage 0.5880845768109223
Origin: Bronx , Destination: Queens , Percentage 0.5657568843718354
Origin: Brooklyn , Destination: Staten Island , Percentage 0.15641585632190919
Origin: Staten Island , Destination: Brooklyn , Percentage 0.10602603677358745
```

عکس 10 - درصد سفرها بین مناطق عدم یکسان

```
... Origin: Manhattan , Destination: Manhattan , Percentage 44.01813057429753
Origin: Queens , Destination: Queens , Percentage 16.80506582398946
Origin: Brooklyn , Destination: Brooklyn , Percentage 14.165273727748565
Origin: Bronx , Destination: Bronx , Percentage 4.920510974731885
```

عکس 11 - درصد سفرها بین مناطق یکسان

بر اساس ماتریس OD، تحلیل جریان سفر بین مناطق نشان می‌دهد که منتهن به عنوان نقطه مرکزی بازار، مقصد اصلی برای بسیاری از سفرهاست و حجم بالای سفرها، چه به صورت ورودی و چه خروجی، اهمیت آن را در حمل و نقل نشان می‌دهد. بروکلین نیز به دلیل تراکم جمعیتی و تقاضای بالای حمل و نقل، در جایگاه دوم اهمیت قرار دارد. بنابراین، ورود به این بازار نیازمند تمرکز اولیه بر این دو منطقه است.

به عنوان تأمین کننده جدید، می توان خدمات خود را به گونه ای طراحی کرد که ابتدا در منهن و بروکلین، با افزایش تعداد خودروها و زمان بندی بهینه برای سرویس دهی در ساعات پرتدد، حضوری قدرتمند ایجاد کرد. این مناطق به دلیل جریان سفرهای مکرر، پتانسیل بالایی برای سودآوری دارند. همچنین می توان از مدل قیمت گذاری پویا استفاده کرد تا در ساعات اوج تقاضا، سود بیشتری حاصل شود.

از طرف دیگر، مناطقی مانند استیتن آیلند و فرودگاه (EWR)، با وجود جریان سفر کم، هنوز پتانسیل جذب مشتریانی خاص را دارند. ارائه خدمات ویژه، مانند سفرهای اشتراکی یا تخفیف برای سفرهای طولانی تر، می تواند به افزایش تقاضا در این مناطق کمک کند. در مناطقی با سفرهای محلی، مانند برونکس و کوئینز، تمرکز بر خدمات سریع و ارزان می تواند نیاز مشتریان را به خوبی برآورده سازد.

به طور کلی، موفقیت در این بازار مستلزم استفاده از داده های ماتریس OD برای شناسایی الگوهای جریان سفر و تطبیق خدمات با نیازهای مناطق مختلف است. تمرکز اولیه بر مناطقی با جریان بالای سفر و درعین حال بهره گیری از فرصت ها در مناطق کم تردد، می تواند به ایجاد مزیت رقابتی کمک کند.

خواسته دوم:

ابتدا تقاضای روزانه سفر را باید محاسبه کنیم. برای این کار اول با استفاده از groupby، تعداد سفر از هر مبدا به هر مقصد در هر تاریخ را به دست می آوریم. با استفاده از فاکتور بالانس، تعداد سفرها را بالانس می کنیم. سپس از این داده بر حسب مبدا و مقصد میانگین گرفته و متوسط تقاضای روزانه را به دست می آوریم.

جدول میانگین تقاضای روزانه را مرتب کرده و مجموع سطری و ستونی آن را حساب می‌کنیم. با استفاده از امتیاز دهی به هر لوکیشن، 25 لوکیشن برتر را انتخاب می‌کنیم. امتیازدهی برای رده‌بندی لوکیشن‌ها بر اساس تعداد کل سفر و تعداد خانه‌های بزرگ‌تر از 0 در جدول انجام شد.

حال تقاضای سفر 25 لوکیشن برتر را جدا می‌کنیم.

...	74	75	41	42	244	95	166	76	7	61	...	55	188	14	39	25	213	35	197	89	226
74	29.708473	53.601002	26.568292	23.478125	7.880976	1.303943	20.266495	1.354095	1.732246	1.404247	...	1.303943	1.303943	1.303943	1.303943	1.385440	1.624755	1.303943	1.371975	1.369141	1.379171
75	55.540841	28.625855	17.433544	14.875671	3.637753	1.633676	7.819987	1.412605	1.691175	1.425240	...	1.459460	1.363213	1.303943	1.356101	1.434338	1.661320	1.303943	1.393870	1.385440	1.462366
41	21.052953	12.810797	22.915730	26.825508	3.985565	1.441201	10.876971	1.303943	1.470404	1.470404	...	1.303943	1.358274	1.564732	1.404247	1.412605	1.405814	1.303943	1.303943	1.303943	1.607186
42	10.186607	6.802544	15.238942	20.855929	4.522438	1.303943	4.690470	1.303943	1.448826	1.348907	...	1.422484	1.363213	1.564732	1.303943	1.303943	1.593709	1.380646	1.303943	1.404247	1.564732
244	2.819835	2.392453	2.674189	4.508816	8.028411	1.341199	5.149800	1.461997	1.303943	1.303943	...	1.303943	1.303943	1.380646	1.303943	1.303943	1.448826	1.303943	1.357166	1.422484	1.303943
95	1.303943	1.338258	1.303943	1.434338	1.360637	24.814328	1.390873	1.490221	1.754571	1.355078	...	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.376385	3.170953	1.303943	1.945781
166	7.849014	4.489526	8.192563	6.901717	5.094123	1.303943	10.935834	1.303943	1.388069	1.303943	...	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943
76	1.382970	1.521267	1.422484	1.385440	1.504550	1.376385	1.629929	3.072718	1.303943	1.804447	...	1.405814	1.801501	1.380646	1.659564	1.412605	1.521267	1.833267	1.352238	1.452119	1.303943
7	1.493225	1.427289	1.378454	1.434338	1.408259	1.609372	1.372572	1.422484	14.606327	1.400532	...	1.412605	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.376385	1.303943	3.793673
61	1.303943	1.376385	1.352238	1.422484	1.303943	1.303943	1.303943	1.918659	1.303943	3.536327	...	1.399354	2.019009	1.420714	1.899711	1.664149	1.303943	1.652325	1.303943	1.535755	1.303943
129	1.356101	1.409668	1.303943	1.372572	1.397082	1.791505	1.376385	1.434338	2.972146	1.629929	...	1.303943	1.303943	1.303943	1.303943	1.434338	1.303943	1.303943	1.466936	1.404247	2.572486
82	1.356101	1.509829	1.303943	1.303943	1.303943	3.151597	1.303943	1.393870	3.171455	1.356101	...	1.303943	1.303943	1.303943	1.303943	1.434338	1.338258	1.613354	1.303943	2.667157	1.303943
130	1.303943	1.326820	1.303943	1.460417	1.303943	1.798871	1.303943	1.472973	1.453576	1.517316	...	1.372572	1.369141	1.303943	1.527476	1.593709	1.303943	1.377405	2.281901	1.303943	1.484489
97	1.303943	1.393870	1.303943	1.376385	1.303943	1.303943	1.422484	1.601988	1.344692	5.878795	...	1.400532	2.870552	1.454926	1.392346	2.728622	1.303943	1.735169	1.388069	2.233003	1.408259
65	1.303943	1.460417	1.434338	1.303943	1.303943	1.369141	1.303943	1.541024	1.376385	5.043624	...	1.399354	2.655069	1.623537	1.428128	2.422388	1.303943	1.553635	1.303943	2.072802	1.303943
55	1.303943	1.465812	1.303943	1.303943	1.303943	1.303943	1.303943	1.571877	1.303943	1.303943	...	4.432264	1.358274	1.446562	1.568512	1.388069	1.469900	1.303943	1.408259	1.532133	1.303943
188	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.872329	1.303943	1.885319	...	1.340164	2.365293	1.346006	1.554702	1.511389	1.621119	1.799112	1.303943	1.814654	1.303943
14	1.417330	1.303943	1.303943	1.303943	1.303943	1.303943	0.000000	1.401739	1.303943	1.397082	...	1.470404	1.303943	2.327037	1.452965	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943
39	1.303943	1.400532	1.303943	1.303943	1.303943	1.303943	0.000000	1.681898	1.303943	1.738591	...	1.494102	1.636321	1.581833	3.190826	1.338258	1.303943	1.457348	1.397082	1.380646	1.303943
25	1.590811	1.303943	1.303943	1.303943	1.448826	1.303943	1.303943	1.525891	1.303943	2.562923	...	1.436548	1.649887	1.476755	1.508483	2.470380	1.303943	1.521267	1.303943	1.778105	1.448826
213	1.481754	1.490221	1.341199	1.517316	1.534051	1.303943	1.466936	1.424307	1.303943	1.376385	...	1.595413	1.327228	1.303943	1.303943	2.298704	1.521267	1.541769	1.303943	1.303943	1.303943
35	1.408259	1.390873	1.303943	1.303943	1.303943	1.303943	1.303943	1.884791	1.466936	1.564732	...	1.335747	1.732872	1.303943	1.564732	1.501511	1.792922	1.787302	1.324478	1.490221	1.303943
197	1.386174	1.425240	1.303943	1.303943	1.438834	1.445676	1.303943	1.452965	1.490221	1.303943	...	1.397082	1.593709	1.303943	1.340164	1.303943	1.534519	1.392561	1.931246	1.303943	1.385440
89	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.554015	1.303943	1.552313	...	1.390873	1.861083	1.303943	1.387174	1.495700	1.324641	1.350513	1.303943	2.258944	1.303943
226	1.303943	1.380646	1.376385	1.348907	1.303943	1.453576	1.303943	1.303943	1.870875	1.303943	...	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.303943	1.448826	1.303943	2.744895

عکس 12 - ماتریس OD لوکیشن‌های برتر

سپس ستون‌های ساعت و روز را بر حسب زمان سوار کردن به داده‌ها اضافه می‌کنیم. تعداد مجموع هر سفر در هر ساعت را به دست می‌آوریم. با توجه به این تعداد در ساعات مختلف با استفاده از k-means ساعات روز را خوشه‌بندی می‌کنیم و مقادیر را به جدول اصلی اضافه می‌کنیم.

hour	Trip_Count	Cluster
0	11521	1
1	7440	1
2	4942	1
3	3881	1
4	4762	1
5	7646	1
6	17580	1
7	32964	2
8	45862	0
9	51730	3
10	53929	3
11	54095	3
12	53826	3
13	52083	3
14	55819	3
15	58624	3
16	56800	3
17	55337	3
18	52155	3
19	42932	0
20	31901	2
21	24992	2
22	21097	2
23	17692	1

عکس 13 - خوشه‌بندی ساعات روز

با توجه به خوشه‌بندی انجام شده و ستون روز هفته در جدول اصلی درصد سفرهای در هر روز هفته و در هر گروه زمانی از روز را محاسبه می‌کنیم.

```
... Time_Group
0    0.114281
1    0.083316
2    0.135727
3    0.666676
dtype: float64

day
0    0.140195
1    0.149129
2    0.157174
3    0.158424
4    0.164526
5    0.133451
6    0.097102
dtype: float64
```

عکس 14 - درصد سفرها در هر گروه زمانی و هر روز هفته

در اینجا با استفاده از لیست 25 لوکیشن برتر، داده‌های مربوط به این لوکیشن‌ها در جدول اصلی را جدا می‌کنیم و متوسط هزینه با توجه به روز هفته و گروه زمانی هر مسیر بین لوکیشن‌های برتر را محاسبه می‌کنیم. بین تعدادی از این مسیرها، سفری در داده‌های جدول اصلی وجود نداشت. هزینه این مسیرها را بی‌نهایت قرار می‌دهیم.

...	PULocationID	DOLocationID	Time_Group	day	Amount
0	74	74	1	4	9.006731
1	74	74	1	5	10.479670
2	74	74	1	6	11.919455
3	74	74	1	0	9.358852
4	74	74	1	1	8.603731
...
17495	226	226	3	6	17.870789
17496	226	226	3	0	14.207077
17497	226	226	3	1	14.948333
17498	226	226	3	2	12.426585
17499	226	226	3	3	16.786508

عکس 15 - متوسط هزینه سفر در هر گروه زمانی روز و هر روز هفته و هر مسیر

در جدولی دیگر، با استفاده از داده فیلتر شده برای لوکیشن‌های برتر، میانگین مسافت میان آن‌ها را محاسبه می‌کنیم و مانند متوسط هزینه مقادیر خالی این جدول را بی‌نهایت می‌گذاریم.

...	PULocationID	DOLocationID	Dist
0	74	74	7.791831
1	74	75	1.345465
2	74	41	16.018143
3	74	42	51.463034
4	74	244	3.505723
...
620	226	213	10.490000
621	226	35	9.852222
622	226	197	9.221500
623	226	89	1987.430000
624	226	226	1.391308

عکس 16 - متوسط مسافت سفر در هر هر مسیر

سپس برای محاسبه مدل جاذبه، تابع بازدارندگی نمایی و لاگ نرمال را با پارامترهای داده شده می‌سازیم. در قدم بعدی تابع مدل جاذبه را تعریف می‌کنیم.

بار دیگر مانند قبل، ماتریس هزینه متوسط را بدون فیلتر کردن روز هفته و گروه زمانی، به دست می‌آوریم. سپس با استفاده از این ماتریس و تابع بازدارندگی‌های مختلف، R-Squared و R-MSE مدل جاذبه با تابع بازدارندگی‌های مختلف را محاسبه می‌کنیم.

```
... RMSE (Expo): 5.458092909503267, R^2 (Expo): -0.49034012223016266  
RMSE (LN): 3.3389301833761857, R^2 (LN): 0.44227742802287373
```

عکس 18 - R-Squared و RMSE توابع بازدارندگی

مدل LogNormal و واضحاً از مدل Exponential هم در R-Squared و هم در شاخص خطا یعنی R-MSE بهتر عمل می‌کند. منفی بودن شاخص R-Squared در مدل نمایی، نشان می‌دهد که حتی در مقایسه با میانگین، برای پیشبینی بدتر عمل می‌کند.

برای پیشبینی پرداخت یا عدم پرداخت انعام، از مدل ترکیبی توسعه داده شده در فاز قبل استفاده می‌کنیم. این مدل برای پیشبینی از Stacking model استفاده کرده که یک روش ترکیبی است. در این متد Base Model روش‌های درخت تصمیم و XGBoost و Metamodel آن، Logistics Regression است و سپس آن‌ها را با Stacking Model ترکیب می‌کند. برای Train و Test مدل از داده‌های جدول اصلی استفاده کرده و خطای آن را نیز محاسبه می‌کنیم.

جدول‌های به دست آمده از متوسط هزینه سفر و مسافت را ادغام کرده و یک جدول تجمیعی از متوسط هزینه برای هر روز هفته، هر گروه زمانی و هر مبدا و مقصد و متوسط مسافت نیز برای هر مبدا و مقصد

را به دست می‌آوریم. با استفاده از مدل آماده شده، پرداخت یا عدم پرداخت انعام را برای هر سطر جدول تجمیعی پیشبینی می‌کنیم.

...	PULocationID	DOLocationID	trip_distance	Time_Group	day	total_amount
0	74	74	7.791831	1	4	9.006731
1	74	74	7.791831	1	5	10.479670
2	74	74	7.791831	1	6	11.919455
3	74	74	7.791831	1	0	9.358852
4	74	74	7.791831	1	1	8.603731
...
17495	226	226	1.391308	3	6	17.870789
17496	226	226	1.391308	3	0	14.207077
17497	226	226	1.391308	3	1	14.948333
17498	226	226	1.391308	3	2	12.426585
17499	226	226	1.391308	3	3	16.786508

عکس 19 - داده‌های تجمیعی مسیرها بین لوکیشن‌ها در هر گروه زمانی و هر روز هفته

در بخش بعدی به سراغ مدل‌سازی می‌رویم. ابتدا تنها متغیر مدل را تعریف می‌کنیم. این متغیر x_{ijdt} است که تعداد سفر از i به j در زمان t و روز d است. i نشان دهنده لوکیشن مبدا، j نمایانگر لوکیشن مقصد، t بیانگر گروه زمانی و d روز هفته است.

در قدم بعدی با استفاده از جدول تجمیعی اطلاعات مربوط به فاصله لوکیشن‌ها و هزینه هر سفر را به عنوان پارامترهای مسئله استخراج می‌کنیم.

برای تعریف شروط مدل‌سازی ابتدا شرط تقاضا را داریم که تعداد سفر از هر لوکیشن به هر لوکیشن در هر گروه زمانی روز در هر روز هفته باید کوچک‌تر یا مساوی مقدار متوسط تقاضا سفر روزانه ضرب در درصد سفر در هر گروه زمانی ضرب در درصد سفر در هر روز ضرب در 7 باشد.

شرط بعدی مربوط به مجموع مسافت طی شده توسط تاکسی‌ها در هر روز است. این شرط بیان می‌کند که مجموع x_{ijdt} ضرب در مسافت i به j برای هر d ، باید کوچک‌تر یا مساوی هزار کیلومتر باشد.

شرط بعدی مربوط به فعالیت تاکسی‌ها تنها در روزهای کاری هفته است. برای این شرط x_{ijdt} برای هر i ، j و t و d های مربوط به روزهای غیر کاری از هفته باید برابر صفر باشد.

شرط آخر مربوط به این موضوع است که با توجه به قیمت ثابت 40 دلار در هر سفر، مسیرهایی که هزینه کمتر از 40 دلار دارند، تقاضا نخواهند داشت. یعنی x_{ijdt} برای t ، j ، i و d که در جدول تجمیعی هزینه کمتر از 40 دلار دارند باید برابر با صفر باشد.

در نهایت مدل را در Solver گروبی اجرا کرده و نتایج روزانه درآمد، و تعداد سفرهای غیر سفر از هر لوکیشن مبدا به هر لوکیشن مقصد، در هر گروه زمانی از روز و هر روز هفته به شرح زیر خواهد بود:

Objective Value (Revenue): 19061.770476202095

$x[74,95,3,4]: 5.0$
 $x[74,95,3,1]: 4.0$
 $x[74,95,3,2]: 5.0$
 $x[74,95,3,3]: 5.0$
 $x[74,76,3,0]: 2.0$
 $x[74,82,3,3]: 4.0$
 $x[74,97,3,4]: 6.0$
 $x[74,97,3,0]: 5.0$
 $x[74,97,3,1]: 5.0$
 $x[74,97,3,2]: 5.0$
 $x[74,97,3,3]: 5.0$
 $x[74,65,3,4]: 2.0$
 $x[74,65,3,1]: 2.0$
 $x[74,65,3,2]: 2.0$
 $x[74,65,3,3]: 2.0$
 $x[74,188,3,1]: 2.0$
 $x[74,14,3,1]: 4.0$
 $x[74,25,3,4]: 3.0$
 $x[74,25,3,1]: 3.0$

x[74,25,3,2]: 3.0
x[74,197,3,0]: 3.0
x[74,89,3,0]: 4.0
x[75,95,3,4]: 3.0
x[75,95,3,0]: 3.0
x[75,95,3,1]: 3.0
x[75,95,3,2]: 3.0
x[75,95,3,3]: 3.0
x[75,61,3,4]: 4.0
x[75,61,3,0]: 3.0
x[75,61,3,1]: 2.0
x[75,61,3,2]: 4.0
x[75,61,3,3]: 4.0
x[75,65,3,2]: 3.0
x[75,188,3,0]: 3.0
x[75,197,3,0]: 2.0
x[75,89,3,0]: 2.0
x[41,95,3,4]: 2.0
x[41,95,3,1]: 2.0
x[41,95,3,2]: 2.0
x[41,95,3,3]: 2.0
x[41,82,3,2]: 2.0
x[41,130,3,0]: 2.0
x[41,97,3,0]: 2.0
x[41,97,3,3]: 2.0
x[41,65,3,4]: 2.0
x[41,65,3,0]: 2.0
x[41,65,3,1]: 2.0
x[41,65,3,2]: 2.0
x[41,65,3,3]: 2.0
x[41,188,3,0]: 3.0
x[42,95,3,4]: 2.0
x[42,7,3,2]: 3.0
x[42,65,3,4]: 2.0
x[42,65,3,2]: 2.0
x[42,65,3,3]: 2.0
x[95,74,3,4]: 4.0
x[95,74,3,0]: 3.0
x[95,74,3,1]: 4.0
x[95,74,3,2]: 4.0
x[95,74,3,3]: 4.0
x[95,75,3,4]: 3.0
x[95,75,3,0]: 2.0
x[95,75,3,1]: 2.0
x[95,75,3,2]: 2.0
x[95,41,3,4]: 2.0
x[95,41,3,0]: 2.0

x[95,41,3,1]: 2.0
x[95,41,3,3]: 2.0
x[95,42,3,4]: 2.0
x[95,42,3,1]: 2.0
x[95,166,3,0]: 4.0
x[95,166,3,1]: 4.0
x[95,166,3,2]: 2.0
x[95,25,3,4]: 2.0
x[95,25,3,3]: 2.0
x[166,95,3,1]: 3.0
x[166,95,3,2]: 3.0
x[166,129,3,2]: 3.0
x[166,82,3,4]: 2.0
x[166,82,3,0]: 2.0
x[166,82,3,2]: 2.0
x[166,82,3,3]: 2.0
x[166,25,3,3]: 2.0
x[166,226,3,4]: 2.0
x[166,226,3,1]: 2.0
x[7,244,3,2]: 2.0
x[7,244,3,3]: 2.0
x[61,74,3,4]: 2.0
x[61,74,3,1]: 2.0
x[61,74,3,2]: 2.0
x[129,42,3,2]: 2.0
x[129,166,3,4]: 3.0
x[129,166,3,3]: 3.0
x[82,41,3,0]: 2.0
x[82,42,3,1]: 2.0
x[82,166,3,1]: 3.0
x[82,166,3,3]: 3.0
x[130,75,3,0]: 2.0
x[130,42,3,0]: 2.0
x[97,74,3,4]: 5.0
x[97,75,3,4]: 2.0
x[97,75,3,0]: 2.0
x[97,75,3,1]: 2.0
x[97,75,3,3]: 2.0
x[97,41,3,4]: 2.0
x[97,41,3,0]: 2.0
x[97,41,3,1]: 2.0
x[97,41,3,3]: 2.0
x[97,42,3,0]: 2.0
x[97,42,3,1]: 2.0
x[65,74,3,4]: 3.0
x[65,74,3,1]: 2.0
x[65,74,3,2]: 2.0

x[65,75,3,1]: 3.0
x[65,75,3,2]: 3.0
x[65,41,3,0]: 2.0
x[65,41,3,2]: 3.0
x[65,41,3,3]: 3.0
x[65,42,3,4]: 3.0
x[55,74,3,3]: 2.0
x[188,74,3,4]: 2.0
x[188,74,3,2]: 2.0
x[25,74,3,4]: 2.0
x[25,74,3,0]: 2.0
x[25,74,3,1]: 2.0
x[25,74,3,2]: 2.0
x[25,74,3,3]: 2.0
x[25,95,3,2]: 2.0
x[25,166,3,3]: 2.0
x[89,74,3,1]: 3.0
x[89,74,3,2]: 4.0

با فرض این که یک سال برابر با 52.1429 هفته است، مجموع درآمد سالانه تقریباً برابر خواهد بود با 1,027,004 دلار.