Industrial Engineering
Department
Sharif University of
Technology

Project Phase 1

Transportation Planning

DR. Hassannayebi

Amirali Modir 400103635
Amirhossein Monji 400103679

قبل از شروع خواسته‌های پروژه، ابتدا کتابخانه‌های لازم فراخوانی شدند. این کتابخانه‌ها عبارت‌اند از:

1. Pandas
2. Numpy
3. Seaborn
4. Matplotlib
5. Sklearn
6. Xgboost
7. Scipy

حالا که ابزارهای مورد نیاز را فراخوانی کردیم، به سراغ خواسته‌های پروژه می‌رویم:

خواسته اول:

اولین قدم Load کردن داده‌هاست. پس از این کار، سوتون‌هایی که به آن‌ها نیازی نداریم را حذف می‌کنیم. سپس برای هر سوتون 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 سوتون payment_type می‌رویم. مشاهده می‌کنیم که در این سوتون 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
```

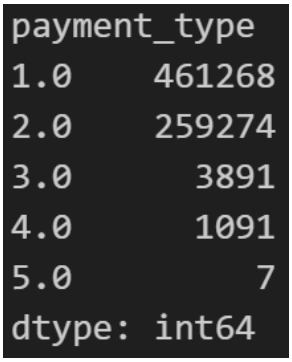
عکس 2 - انواع پرداخت، هنگامی که انعام پرداخت شده باشد.

حال تعداد مسافرهای هر سفر را بررسی می‌کنیم. اگر این عدد بیشتر از 4 بود، ماشین حتما 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.



عکس 4 - انواع پرداخت در کل داده‌ها

حالا 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 dropoff_datetime و pickup_datetime را تصحیح کرده به Datetime تغییر دهیم و سپس فیلتر را انجام دهیم.

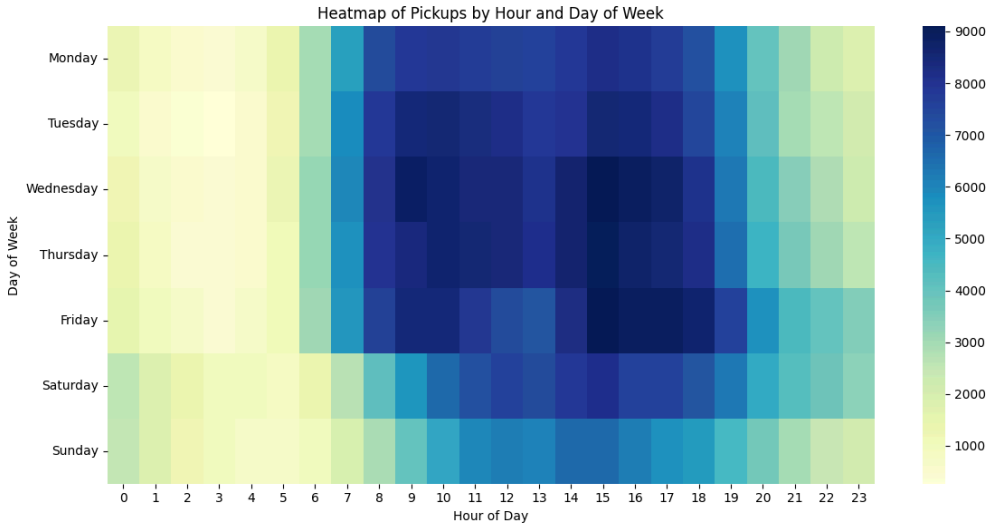
بعد از انجام تمام این کار ها، همچنان Dataset در بعضی نقاط Missing دارد.پس با استفاده از RandomForest ابتدا Feutre های اثرگذار بر آن‌ها را انتخاب کرده (Feature 4 بیشترین اثرگذار) و داده‌های خالی را Predict می‌کنیم. پس از انجام این کار، خواسته اول سوال تمام شده و داده ما پاکسازی شده. حالا به سراغ خواسته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

```
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
```

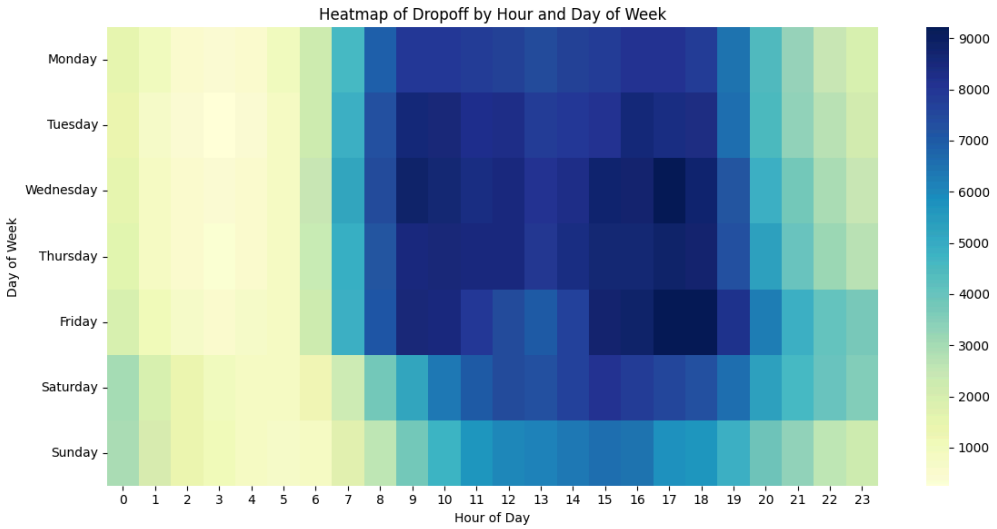
عکس 5 - نسبت missing در هر ستون قبل از پیشبینی

خواسته دوم:

برای رسم نمودار نقشه گرمایی، ابتدا باید ساعت و روز را به Dataset اضافه کنیم. ستون‌های اضافه شده به Dataset، که Day و Hour زمان سوار کردن مسافر و همچنین Day2 و Hour2 زمان پیاده کردن مسافر است. حالا بر اساس این دو Heat Map مربوط به سوار شدن و پیاده شدن را رسم می‌کنیم.



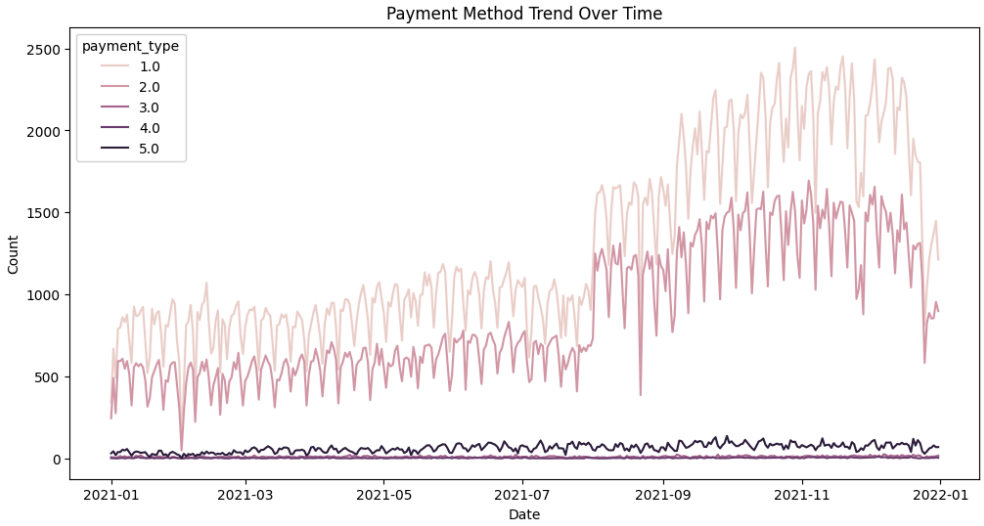
نمودار 1 - Heatmap سوار شدن مسافران به تاکسی



نمودار 2 - Heatmap پیاده شدن مسافران به تاکسی

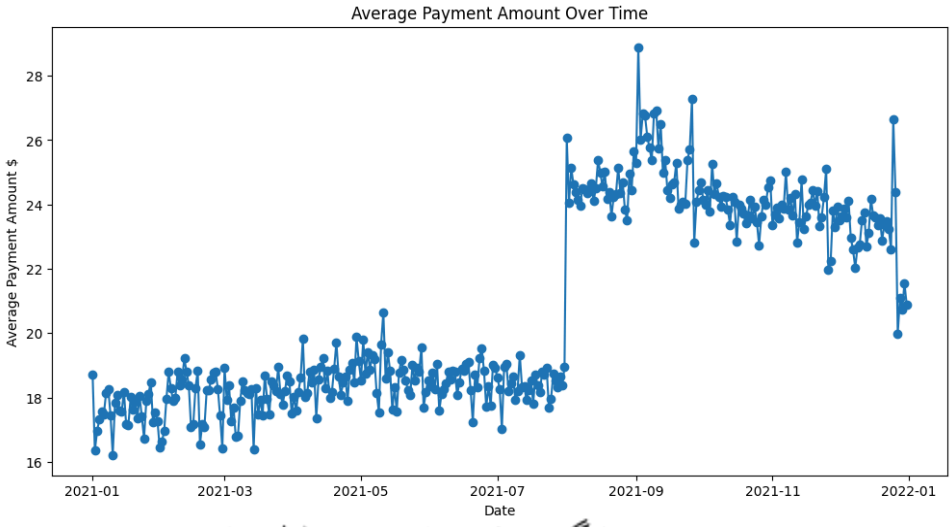
می‌بینیم که دو نمودار تشابه زیادی با یکدیگر دارند و این موضوع منطقی نیز هست. در هر دو مشاهده می‌کنیم که خلوت‌ترین روز ها روزهای تعطیل یعنی شنبه و یکشنبه هستند. همچنین اوج شلوغی در روزهای هفته معمولا از 8 صبح شروع شده و تا ساعت 6 ادامه دارد.

حالا به سراغ بررسی روند پرداخت مسافران در گذر زمان، روند در نوع سفر، روند در استفاده از تاکسی سبز در هر یک از بخش‌های شهر و روند استفاده در بازه‌های زمانی روز می‌رویم.



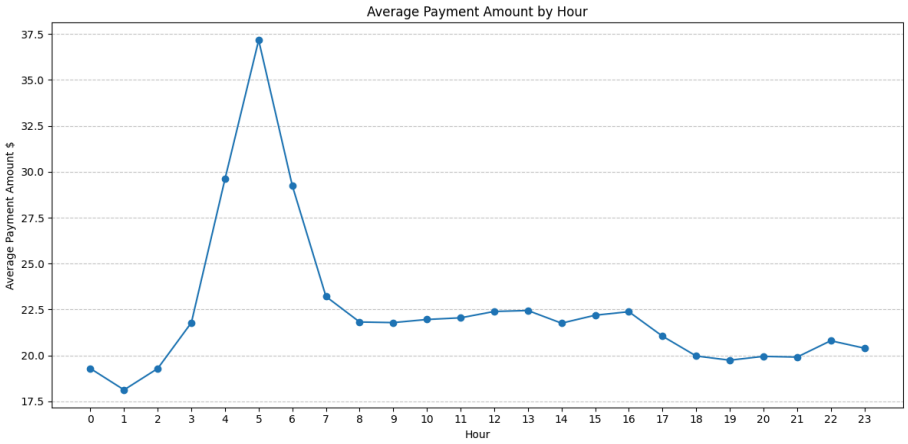
نمودار 3 - نحوه پرداخت در طول زمان

ابتدا نمودار نحوه پرداخت در طول زمان را رسم می‌کنیم. مشاهده می‌کنیم که نوع اول و دوم بسیار بیشتر از سایر موارد هستند. همچنین تلورانس نسبتا بالایی دارند و در گذر زمان نیز روند صعودی داشته‌اند. اما نحوه پرداخت 3، 4 و 5 هر سه بسیار کم بوده‌اند. نه رو به افزایش‌اند و نه کاهش و تقریبا در طول زمان ثابت بوده‌اند و همچنین تلورانس کمی نیز داشته‌اند.



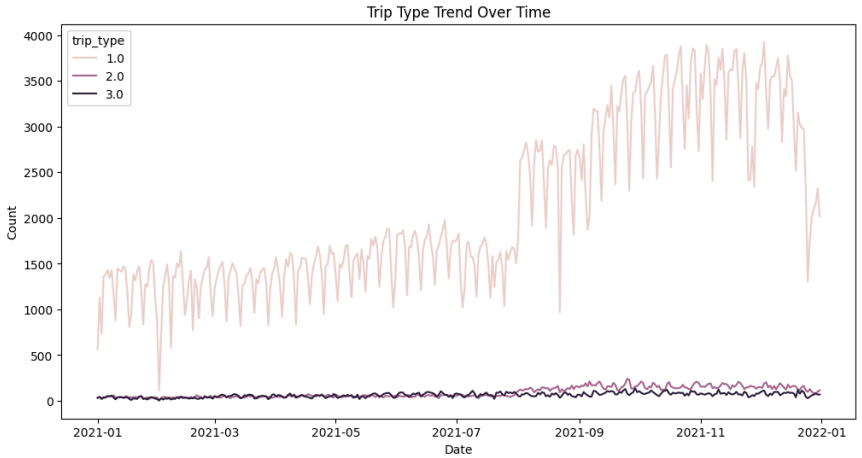
نمودار 4 - میانگین مبلغ پرداختی در طول زمان

سپس نمودار میانگین مبلغ پرداختی را در طول زمان می‌کشیم. مشاهده می‌کنیم که قبل از ماه هشتم، روند صعودی بسیار ملایمی داشته با تلورانس زیاد. سپس در ماه هشتم شاهد پرش بسیار بزرگی در متوسط پرداختی بودیم. این موضوع می‌توان مربوط به افزایش قیمت‌ها بوده باشد. سپس بعد از ماه هشتم، شاهد روند نزولی ملایمی هستیم که تلورانس نسبتا بالایی نیز دارد. روند کاهشی می‌تواند به علت افزایش قیمت‌ها و جایگزین کردن نوع سفر توسط مردم به علت قیمت بالای آن بوده باشد.



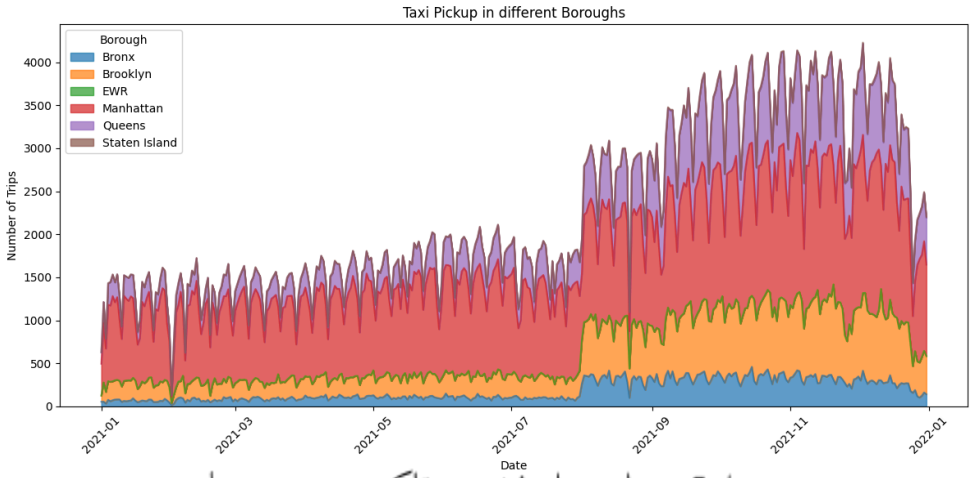
نمودار 5 - میانگین مبلغ پرداختی در طول روز

نمودار بعدی مربوط به میانگین پرداخت بر اساس ساعت شبانه روز بوده. مشاهده می‌کنیم که بیشترین زمان، مربوط به ساعت 5 صبح بوده. این می‌تواند به علت قیمت بسیار بالای سفر در این ساعت شبانه روز بوده باشد. درخواست تاکسی و حجم ترافیک به علت رفتن افراد به سرکار در این ساعت بیشتر است.

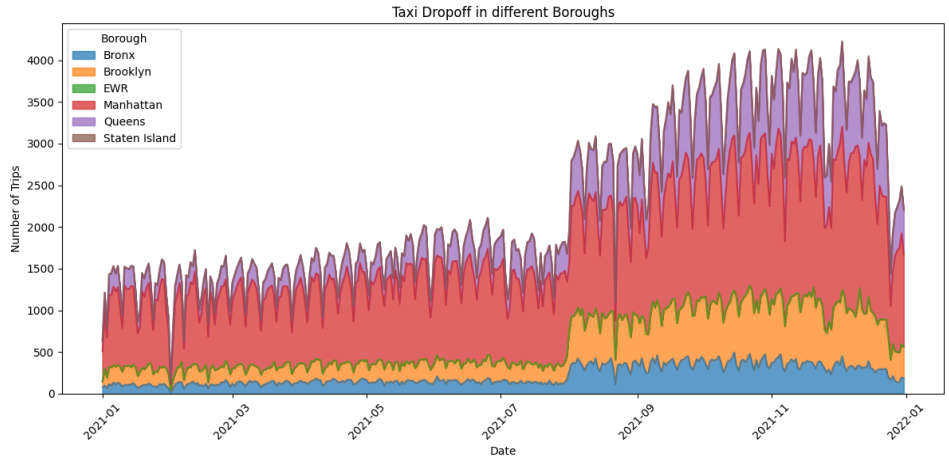


نمودار 6 - نوع سفر در طول زمان

حال به سراغ بررسی روند نوع سفر می‌رویم. نمودار بعدی نوع سفر بر اساس زمان است. مشاهده می‌کنیم که نوع اول از سایر بسیار بیشتر بوده، روند نسبتا صعودی داشته و تلورانس بالایی نیز داشته. اما نوع دوم و سوم بسیار کمتر از نوع اول و همچنین تقریبا ثابت بوده‌اند و تلورانس بسیار کمی نیز داشته‌اند.

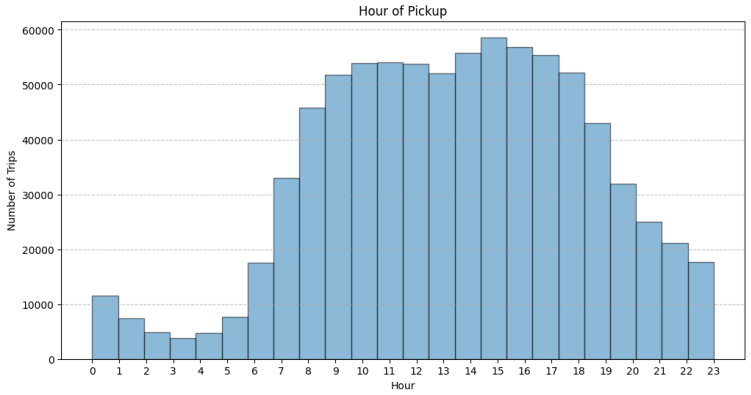


نمودار 7 - میزان سوار شدن به تاکسی در هر محله

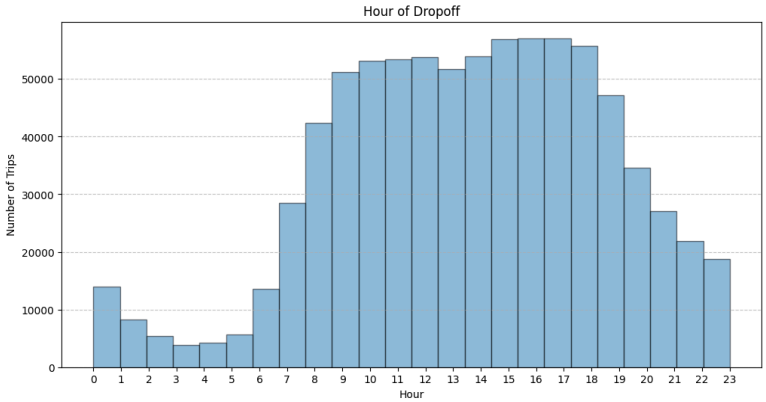


نمودار 8 - میزان پیاده شدن از تاکسی در هر محله

نمودارهای بعدی مربوط به تعداد سوار شدن و پیاده شدن از تاکسی در محله‌های مختلف در طول سال 2021 است. طبق نمودارها مشاهده می‌کنیم که بیشترین سوار و پیدا شدن مربوط به محله Manhattan بوده. همچنین می‌بینیم که سه محله Bronx، Brooklyn و Queens در ماه هشتم افزایش چشمگیری داشته‌اند. موضوع دیگر این است که محله‌های Staten Island و EWR به نسبت سایر محله‌ها، حجم سفر بسیار کمتری داشته‌اند.



نمودار 9 - تعداد سوار شدن به تاکسی در ساعات مختلف شبانه‌روز

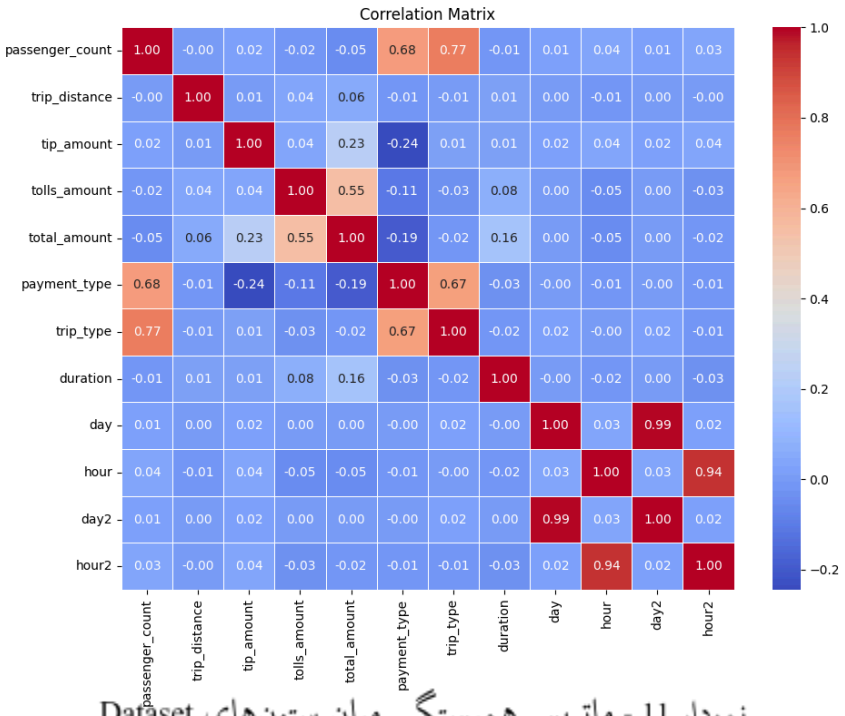


نمودار 10 - تعداد پیاده شدن از تاکسی در ساعات مختلف شبانه‌روز

در نهایت هیستوگرام‌های مربوط به تعداد سفر بر اساس ساعت شبانه‌روز یکی برای ساعت سوار شدن و دیگری برای پیاده شدن رسم شده‌اند. همچنان مشاهده می‌کنیم که بیشتری ساعت سفر، بین بازه 9 صبح تا 6 عصر است. جدای از خواسته‌های مساله، ساعت پیک سوار شدن و میانگین آن محاسبه شده‌اند. میانگین آن حدود ساعت 13:43 و پیک آن ساعت 15 بوده.

خواسته سوم:

ابتدا داده‌های پرت مربوط به trip_distance با استفاده از z_score حذف شدند تا Dataset تمیزتری داشته باشیم. برای کشیدن ماتریس همبستگی، ابتدا سوتون‌هایی که تایپ آن‌ها Int یا Float نیست را حذف می‌کنیم. سپس بر اساس متغیرهای باقی‌مانده، ماتریس همبستگی را می‌کشیم که نتیجه آن به شرح زیر بود:



نمودار 11 - ماتریس همبستگی میان ستون‌های Dataset

همانطور که انتظار داشتیم، ساعت و روز سوار شدن و پیاده شدن با یکدیگر همبستگی بسیاری زیادی دارند. پس در نتیجه تنها باید از یکی از آن‌ها استفاده کنیم.

در سوال بعدی باید بر اساس روش‌های مربع کای، Backward، Forward و همچنین Random Forest تلاش کنیم تا با بهترین حالت Total Amount را پیشبینی کنیم. ابتدا تابع هر کدام از روش‌ها را تشکیل می‌دهیم.

- روش Backward ابتدا از همه متغیرها استفاده می‌کند و سپس یکی یکی از آن‌ها کم می‌کند و ترکیب های مختلف آن‌ها را بررسی می‌کند.
- روش Forward اما ابتدا با تک متغیر شروع کرده و یکی یکی متغیر اضافه می‌کند تا ترکیب بهینه آن‌ها را پیدا کند. دقیقا برعکس روش قبلی.
- روش مربع کای مربوط به داده‌هایی است که می‌توان روی آن‌ها Clustering انجام داد. روی Dataset ما، این روش به خوبی جواب نمی‌دهد و حاصل آن کارا نیست و از دو روش قبلی نتیجه بدتری دارد.
- روش Random Forest همه حالات را امتحان کرده و متغیرهایی که می‌توان با آن‌ها بهترین رگرسیون را تشکیل داد به ما ارائه می‌کند. ایده اصلی پشت این کار، ترکیب درخت های تصمیم گیری متعدد در تعیین خروجی نهایی به جای تکیه بر درخت های تصمیم فردی است. تشکیل مدل RandomForestRegressor به ما می‌تواند لیست میزان اهمیت هر feature را بدهد.

در این پروژه، چون متغیرهای بسیار زیادی داریم، روش Backward کارا نیست. همچنین درباره Chi-Squared نیز توضیح داده شد. پس بهترین روش‌ها، Random Forest و Forward هستند که اولی، نتیجه بهتری داشت. نتایج این روش‌ها به شرح زیر است؛ در این نتایج از روش اول 3 متغییر، روش دوم 5 متغییر، روش سوم سطح معناداری 5 درصد و از روش چهارم نیز 5 متغییر خواسته شده است.

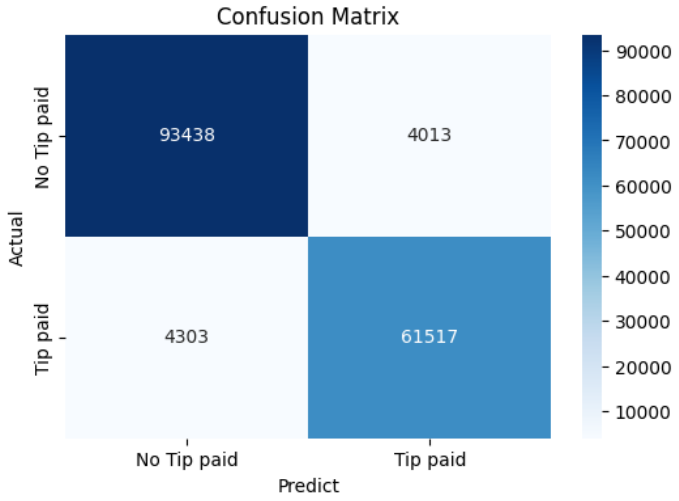
```
Backward: ['passenger_count', 'trip_type', 'day']
Forward: ['tolls_amount', 'tip_amount', 'duration', 'payment_type', 'trip_type']
Chi_square: ['trip_distance', 'tip_amount', 'tolls_amount', 'duration']
           Feature  Importance
1    trip_distance    0.673254
6         duration    0.186945
2       tip_amount    0.051874
8             hour    0.027675
7              day    0.016054
5        trip_type    0.014831
4     payment_type    0.012865
3     tolls_amount    0.011545
0  passenger_count    0.004958
Random Forest: ['trip_distance', 'duration', 'tip_amount', 'hour', 'day']
```

عکس 6 - نتایج روش‌های مختلف انتخاب Feature برای پیش‌بینی total_amount

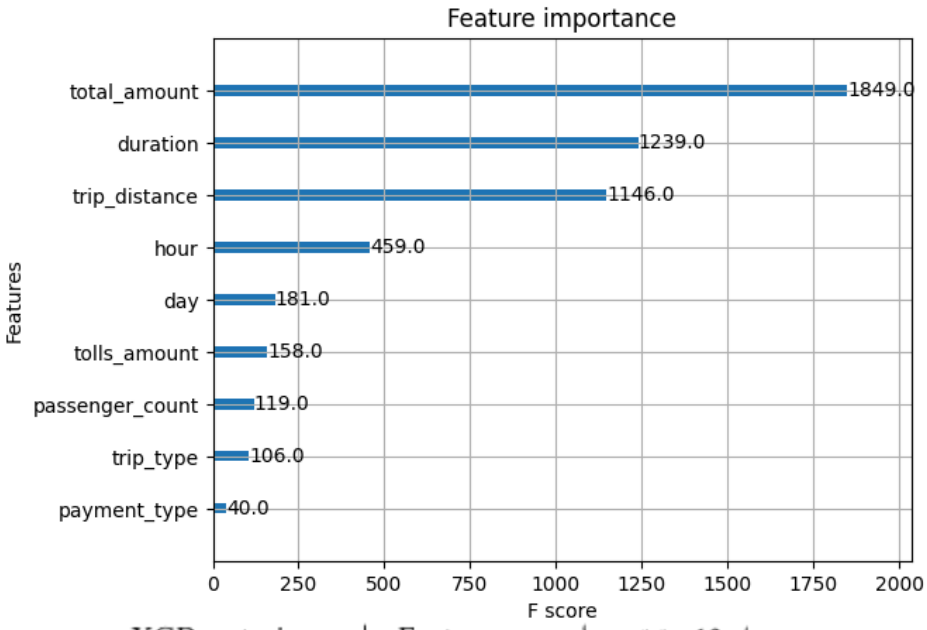
مرحله بعدی، ستون جدیدی اضافه می‌کنیم که یک متغیر باینری است. این متغیر 1 می‌گیرد اگر انعام پرداخت شده باشد و در غیر این صورت صفر خواهد بود. حال می‌خواهیم پرداخت یا عدم پرداخت انعام را پیشبینی کنیم. برای این منظور می‌خواهیم متغیر جدید تعریف شده را بر اساس سایر متغیرها Predict کنیم. ستون tip_amount را نیز در پیش‌بینی نمی‌آوریم تا مدل به صورت صحیح fit شود. قدم اول برای این موضوع جدا کردن داده‌های تست و Train است.

این کار را ابتدا با روش‌های XGBoost, Random Forest، Decision Tree انجام می‌دهیم. سپس به عنوان بخشی امتیازی، از Staking model استفاده کردیم 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 [1].

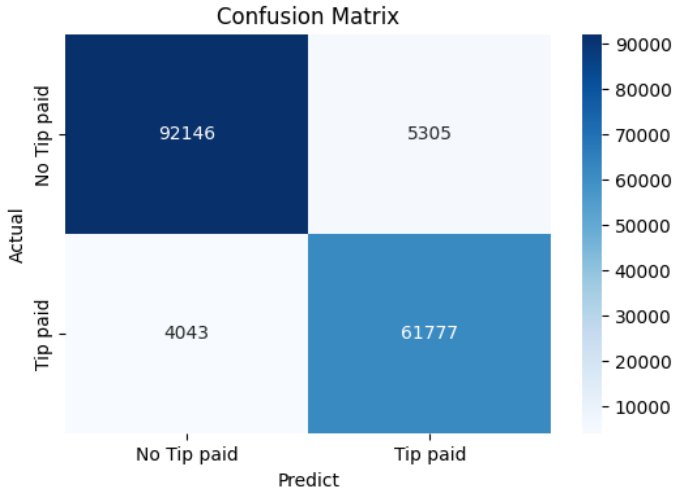
در آخر توابع مربوط به هرکدام از روش ها را اجرا کرده و Confusion Matrix مربوط به آن‌ها را رسم می‌کنیم. در روش XGBoost علاوه بر این ماتریس، میزان اهمیت هر Feature را نیز به ما گزارش می‌کند.



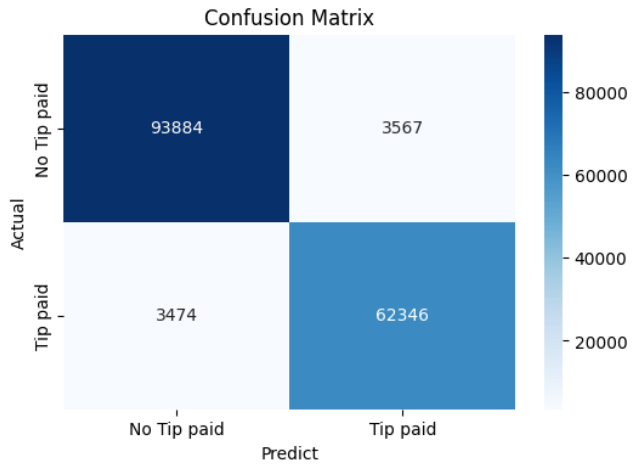
نمودار 12 - ماتریس confusion در روش DesicionTree



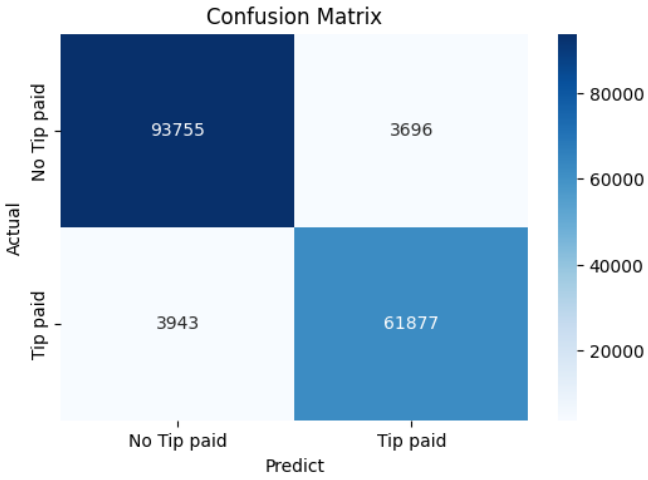
نمودار 13 - ترتیب اهمیت Feature ها در روش XGBoost



نمودار 14 - ماتریس confusion در روش XGBoost



نمودار 15 - ماتریس confusion در روش RandomForest



نمودار 16 - ماتریس confusion در روش Stacking Classifier

حالا به سراغ سوال چهارم می‌رویم. در این سوال خواسته شده تا Total Amount با یک روش ساده و یک روش پیشرفته پیش بینی شود. برای این بخش از تمامی روش‌های ممکن استفاده شد. ابتدا رگرسیون خطی ساده، XGBoost و همچنین Random Forest. همچنین برای بخش امتیازی، از ترکیب Gradient Boosting و Neural Network استفاده شد [2].

در سه روش اول، ابتدا بر حسب لیست اهمیت به دست آمده از روش RandomForest در سوال 2، Feature ها را انتخاب می‌کنیم و سپس داده‌های Test و Train را جدا کرده و بعد مدل Run می‌شود. در رگرسیون خطی ساده، ضرایب رگرسیون را نیز خواهیم داشت.

اما در روش ترکیبی، ابتدا با XGBoost، متغیرهای مهم‌تر را شناسایی می‌کند. سپس با استفاده از شبکه عصبی، یک مدل روی آن فیت می‌کند و R^2 آن را گزارش می‌کند.
ابتدا لیست 7 متغییر مهم را تعریف کرده، سپس متغیرها را یکی یکی اضافه کرده، یک بار با رگرسیون خطی، بار دیگر از XGBoost استفاده کرده و در آخر از RandomForest استفاده می‌کنیم، و نتیجه آن‌ها را پرینت می‌کند. سپس در نهایت Gradient Boosting + Neural Network را ران کرده و نتیجه نهایی را گزارش می‌کند.

نتیجه‌ها نشان می‌دهد که مدل رگرسیون خطی ساده بسیار بد عمل می‌کند و نتایج R^2 آن با تعداد feature های مختلف در حدود 0.004 تا 0.37 است، اما XGBoost مقادیر R^2 بالایی به ما می‌دهد، در حدود 0.74 تا 0.84 و در مورد RandomForest نیز ضعیت به همین صورت است یعنی R^2 بین 0.68 تا 0.87 است؛ اما از لحاظ زمان، XGBoost بسیار بهتر عمل می‌کند و RandomForest بسیار کند است.
نتایج Gradient Boosting + Neural Network نیز به شکل زیر است:

```
Gradient Boosting + Neural Network Results:
Features selected: ['trip_distance', 'tolls_amount', 'payment_type', 'trip_type', 'duration']
MAE: 3.368005500872552
R2 Score: 0.8359696368854873
```

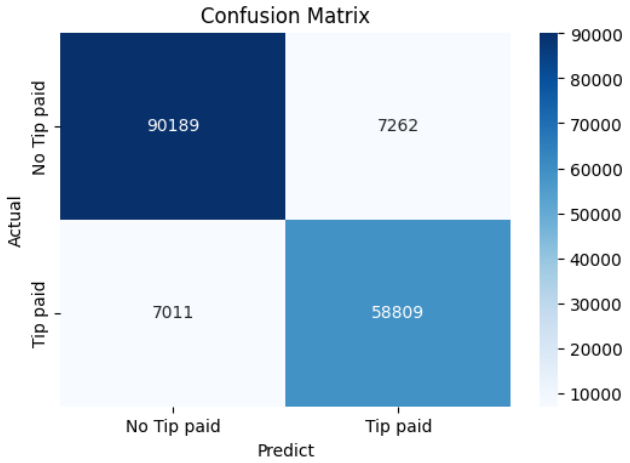
عکس 7 - نتایج روش Gradient Boosting + Neural Network

برای سوال آخر، ابتدا 5 درصد داده‌ها را انتخاب کرده و سپس، چهار عدد از مهم‌ترین متغیرها که توسط XGBoost پیدا شده بودند را شناسایی کرده و سپس Test و Train را جدا می‌کنیم. سپس Valueهای Random Forest را Grid بندی کرده و سپس Experimental Setting آن را ایجاد کرده و سپس حالت‌های مختلف آن را بررسی کرده و کمترین MSE را پیدا می‌کنیم و بهتر حالت را ران می‌گیریم. این کار را هم برای سوال سوم و هم چهارم انجام می‌دهیم. برای سوال چهارم با این تفاوت که متغییرهای مهمی که Gradiant Boosting انتخاب کرده را در داده‌ها انتخاب می‌کنیم.

نتایج آن به شکل زیر است:

```
Best result: N Estimators = 200, Max Depth = 20, Min Samples Split = 2 => MSE = 0.08
```

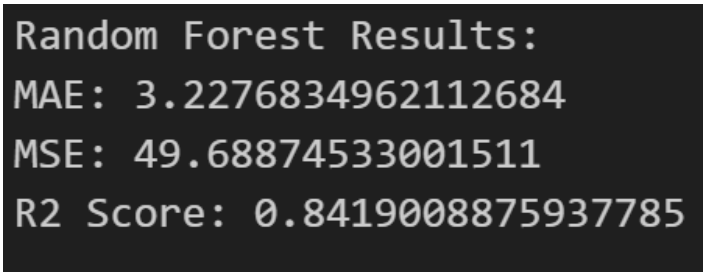
عکس 8 - حالت بهینه سوال 3



نمودار 17 - ماتریس confusion در روش RandomForest با شرایط بهینه

```
Best result: N Estimators = 200, Max Depth = 20, Min Samples Split = 2 => MSE = 0.08
```

عکس 9 - حالت بهینه سوال 4



عکس 10 - نتیجه RandomForest در سوال چهار با شرایط بهینه

منابع:

[1] **Stacked generalization by David H. Wolpert**

[2] **LightGBM：A Highly Efficient Gradient Boosting Decision Tree by Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye and Tie-Yan Liu**