New York City Taxi

Yellow Taxis trip

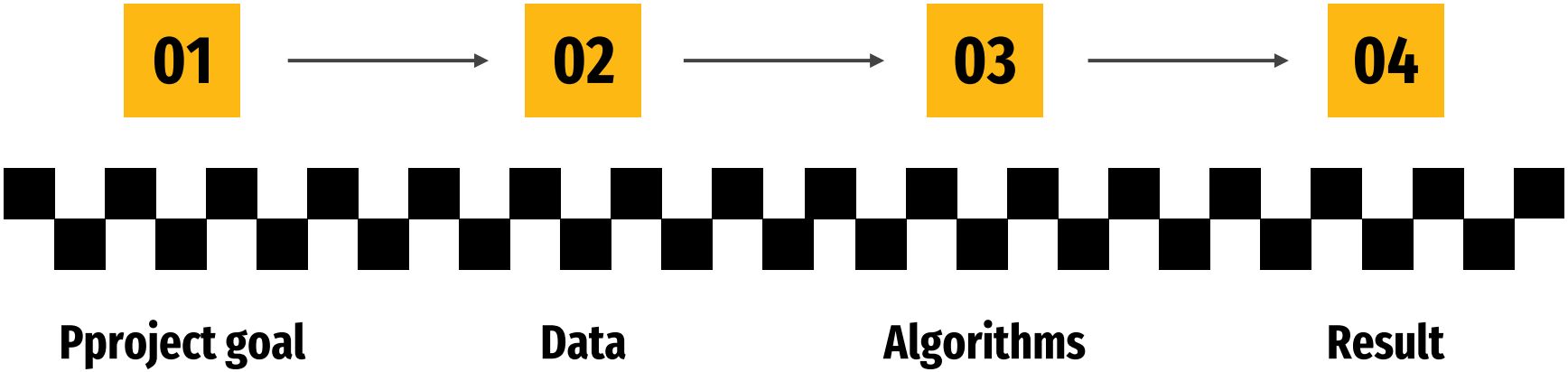
T5 BootCamp Data Science Project

Done by:

Arwa Alolyani **Maitha Alqahtani**

Submit to: Mr.Ali El-kassas

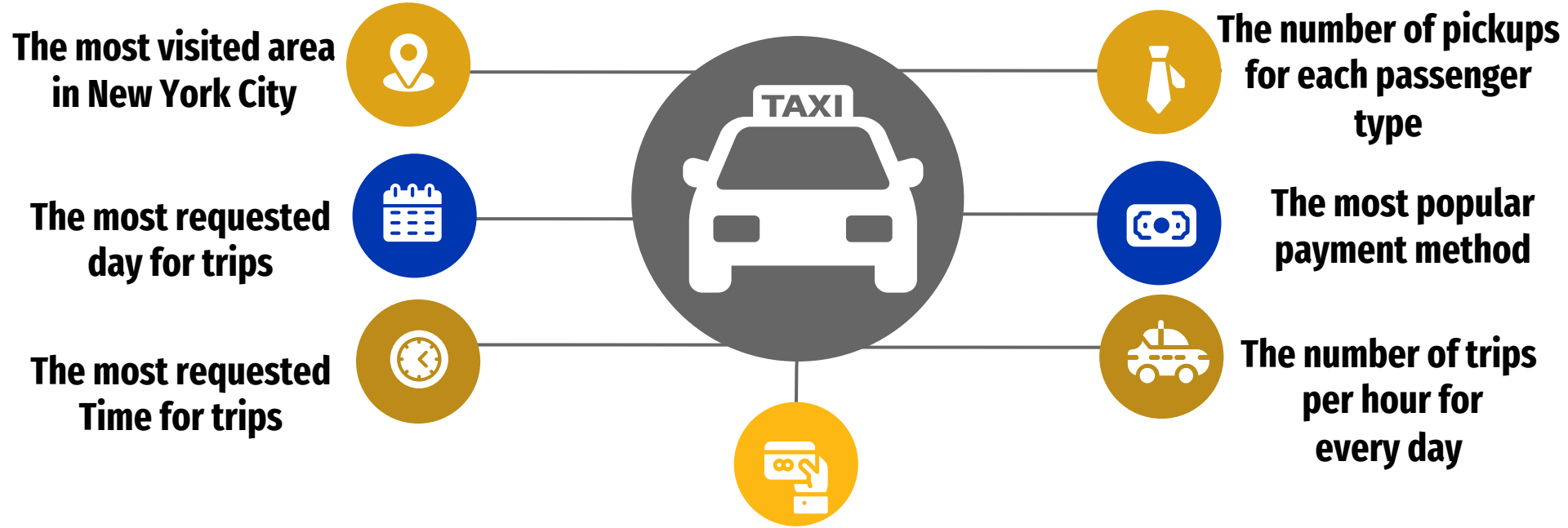
Contents



Abstract

New York City has a huge population that moves from place to place, which means taxi rides are a major part of the traffic. Almost half of the people of New York rely on public transportation, and it is estimated that 54 percent don't own a car or personal vehicle. As a matter of fact, 200 million taxi trips are made annually. The amount of taxi we will need to pay will only be known after we arrive at our destination, and our awkwardness makes us curious about how we can predict this amount, and an accurate estimation will help us control our budget.

Project goal



Data

01

New York City Taxi and Limousine Commission (TLC)

- Pick-up and Drop-of times
- Pick-up and drop-o locations
- Journey distances
- Detailed fares
- Fares types
- Payment types
- Number of passengers

02

NYC Taxi Zones data from NYC OpenData

- shape area
- shape length
- location id
- zone
- borough
- geometry

03

Central Park weather data from the National Climatic Data Center.

- STATION
- NAME
- DATE
- AWND
- PRCP
- SNOW
- SNWD
- TMAX
- TMIN

Algorithms

1 Download and read data

Choosing 1,000,000 as a random sample
Divide the data into 3 sections: Training, Verification and Test

```
df.shape
```

```
(6405008, 18)
```

```
df_sample= df.sample(n = 1000000)
```

```
df_sample
```

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID
3300732	2.0	2020-01-17 16:38:50	2020-01-17 16:46:36	2.0	1.08	1.0	N	230	16
1247601	2.0	2020-01-08 07:25:57	2020-01-08 07:42:03	1.0	2.84	1.0	N	186	12
2157529	2.0	2020-01-12 09:37:04	2020-01-12 09:45:56	1.0	1.48	1.0	N	230	24
3255624	2.0	2020-01-17 12:30:27	2020-01-17 12:43:02	1.0	1.72	1.0	N	48	14
4768513	1.0	2020-01-24 19:16:02	2020-01-24 19:29:52	1.0	2.00	1.0	N	211	18
...	...	...	...	...	...	...	...	...	..
445670	2.0	2020-01-03 17:24:24	2020-01-03 17:34:03	1.0	1.06	1.0	N	186	24
4549219	2.0	2020-01-23 20:11:39	2020-01-23 20:15:11	5.0	1.00	1.0	N	230	4
1552365	1.0	2020-01-09 15:15:40	2020-01-09 15:18:14	1.0	0.40	1.0	N	246	24
587533	1.0	2020-01-04 14:46:27	2020-01-04 14:50:28	3.0	0.70	1.0	N	90	10
4715955	1.0	2020-01-24 15:37:40	2020-01-24 15:50:07	1.0	1.70	1.0	N	236	14

1000000 rows × 18 columns

Algorithms

2 Quick Look at the Data Structure

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000000 entries, 3300732 to 4715955
Data columns (total 18 columns):
 #   Column                 Non-Null Count    Dtype
---  ------                 --------------    -----
 0   VendorID               989893 non-null   float64
 1   tpep_pickup_datetime   1000000 non-null  object
 2   tpep_dropoff_datetime  1000000 non-null  object
 3   passenger_count        989893 non-null   float64
 4   trip_distance          1000000 non-null  float64
 5   RatecodeID             989893 non-null   float64
 6   store_and_fwd_flag     989893 non-null   object
 7   PULocationID           1000000 non-null  int64
 8   DOLocationID           1000000 non-null  int64
 9   payment_type           989893 non-null   float64
 10  fare_amount            1000000 non-null  float64
 11  extra                  1000000 non-null  float64
 12  mta_tax                1000000 non-null  float64
 13  tip_amount             1000000 non-null  float64
 14  tolls_amount           1000000 non-null  float64
 15  improvement_surcharge  1000000 non-null  float64
 16  total_amount           1000000 non-null  float64
 17  congestion_surcharge   1000000 non-null  float64
dtypes: float64(13), int64(2), object(3)
memory usage: 145.0+ MB
```

	VendorID	passenger_count	trip_distance	RatecodeID	PULocationID	DOLocationID	payment_type	fare_amount	extra	5
count	587683.000000	587683.000000	587683.000000	587683.000000	587683.000000	587683.000000	587683.00000	587683.000000	587683.000000	
mean	1.669722	1.514221	2.821691	1.046430	164.294567	162.137365	1.26953	12.283715	1.104000	
std	0.470314	1.149875	3.642314	0.655004	65.158177	69.441014	0.47296	10.883605	1.258006	
min	1.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.00000	-175.000000	-27.000000	
25%	1.000000	1.000000	0.960000	1.000000	132.000000	113.000000	1.00000	6.500000	0.000000	
50%	2.000000	1.000000	1.600000	1.000000	162.000000	162.000000	1.00000	9.000000	0.500000	
75%	2.000000	2.000000	2.880000	1.000000	233.000000	234.000000	2.00000	13.500000	2.500000	
max	2.000000	6.000000	60.410000	99.000000	263.000000	263.000000	4.00000	730.000000	87.560000	

8 rows × 23 columns

Algorithms

3 Prepare the Data for Machine Learning Algorithms

Deleted null values

Deleted all values with less than 0 passengers

Deleted all values less than 2.5 in the fare amount

Deleted all points outside of New York

Deleted all values less than 0 in the distances

Algorithms

4 Feature Engineering

Extract longitude and latitude from geometry columns using geopandas

Calculation of actual distance by latitude and longitude

Change date columns type to date type

Extract the day, hour and time from the date columns

Merging weather data for New York on January 1, 2020

Algorithms

5 Models

Looking for Correlations

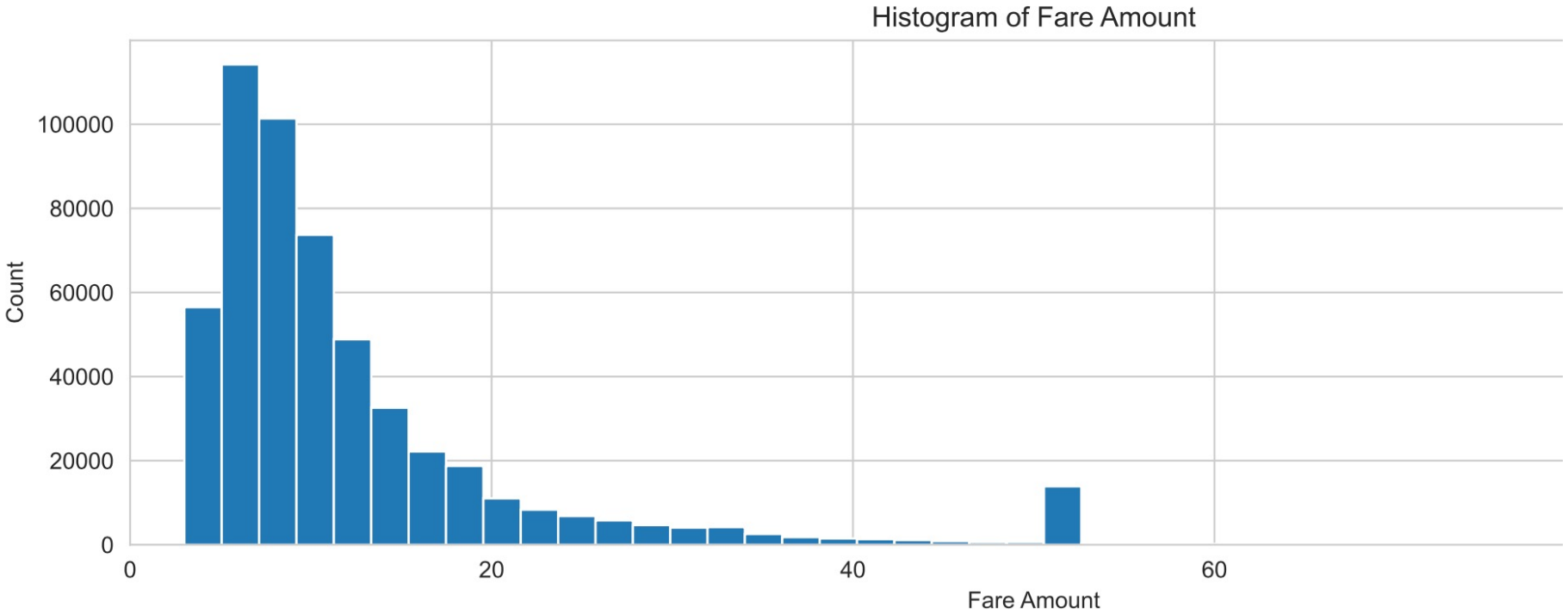
Create linear regression model for predict fare amount

Training and Evaluating on the Training Set

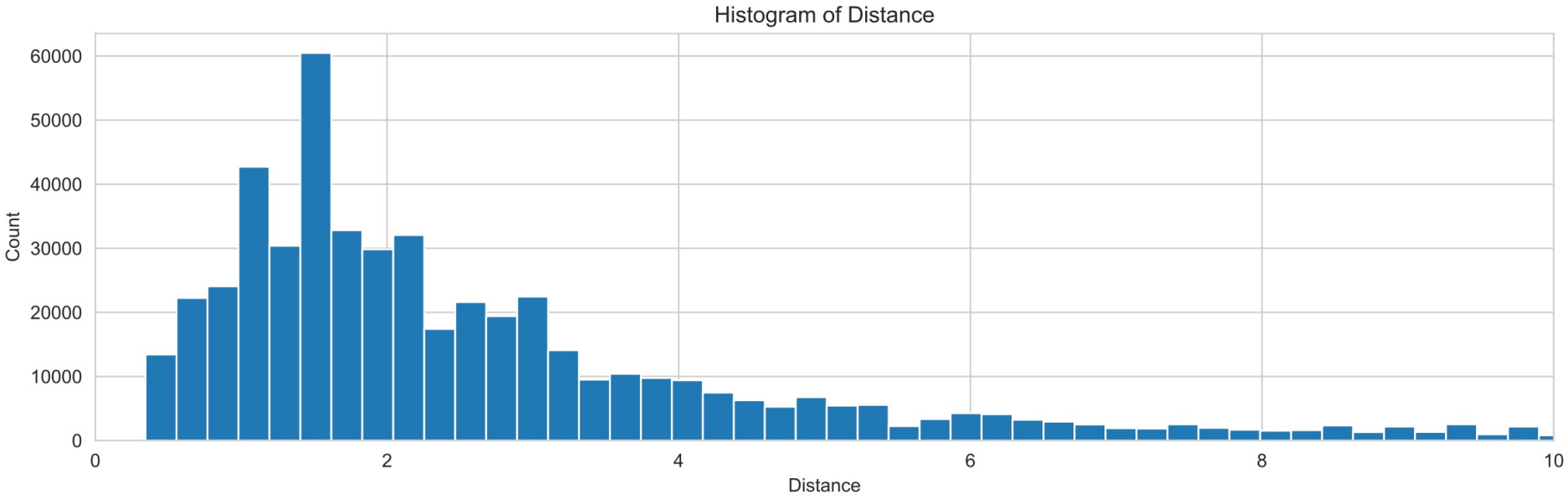
Evaluate the Test Set

Visualizations and Results

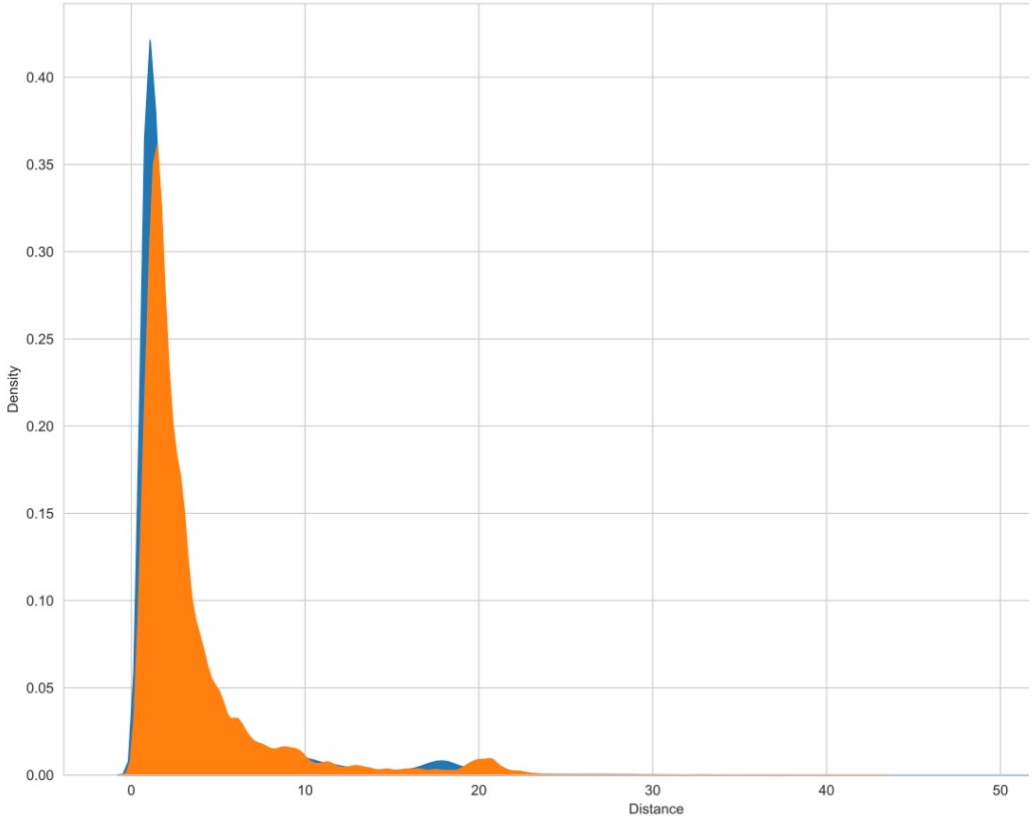
Histogram of Fare Amount



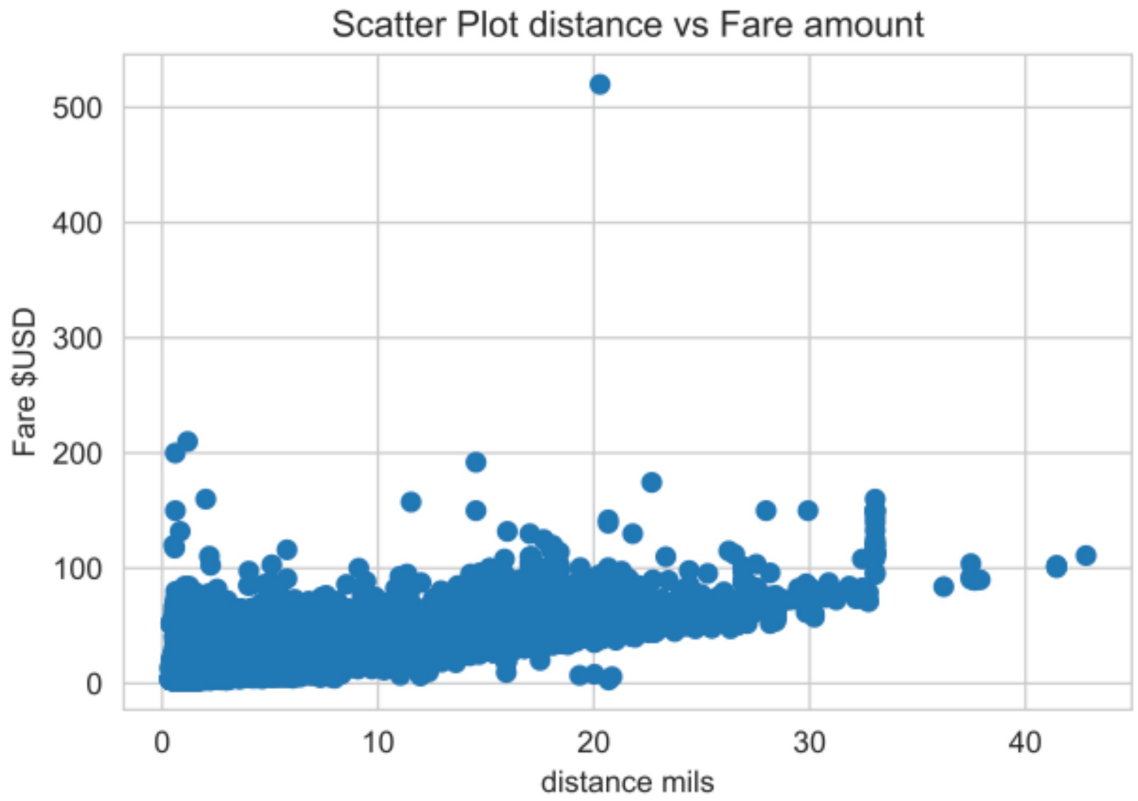
Histogram of Distance



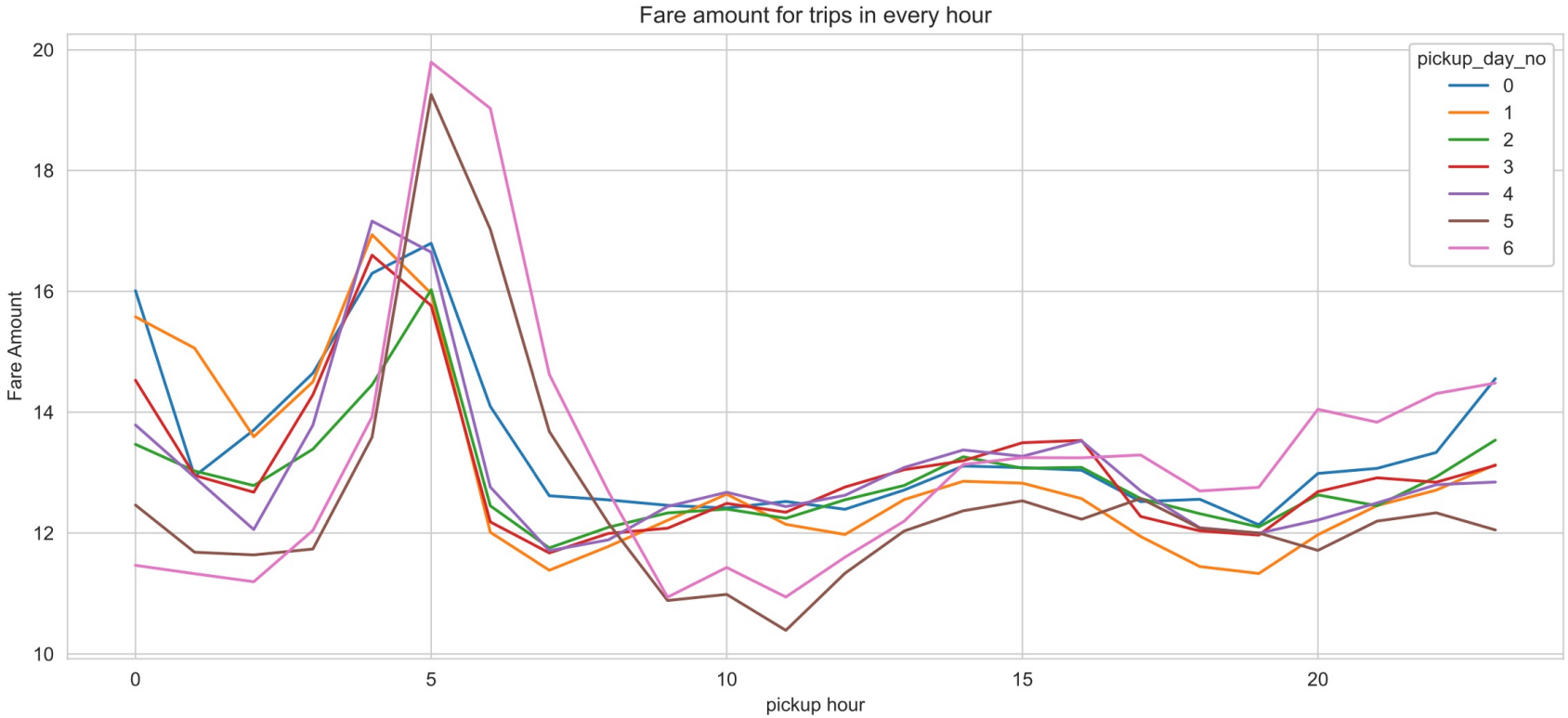
Distances Histogram difference



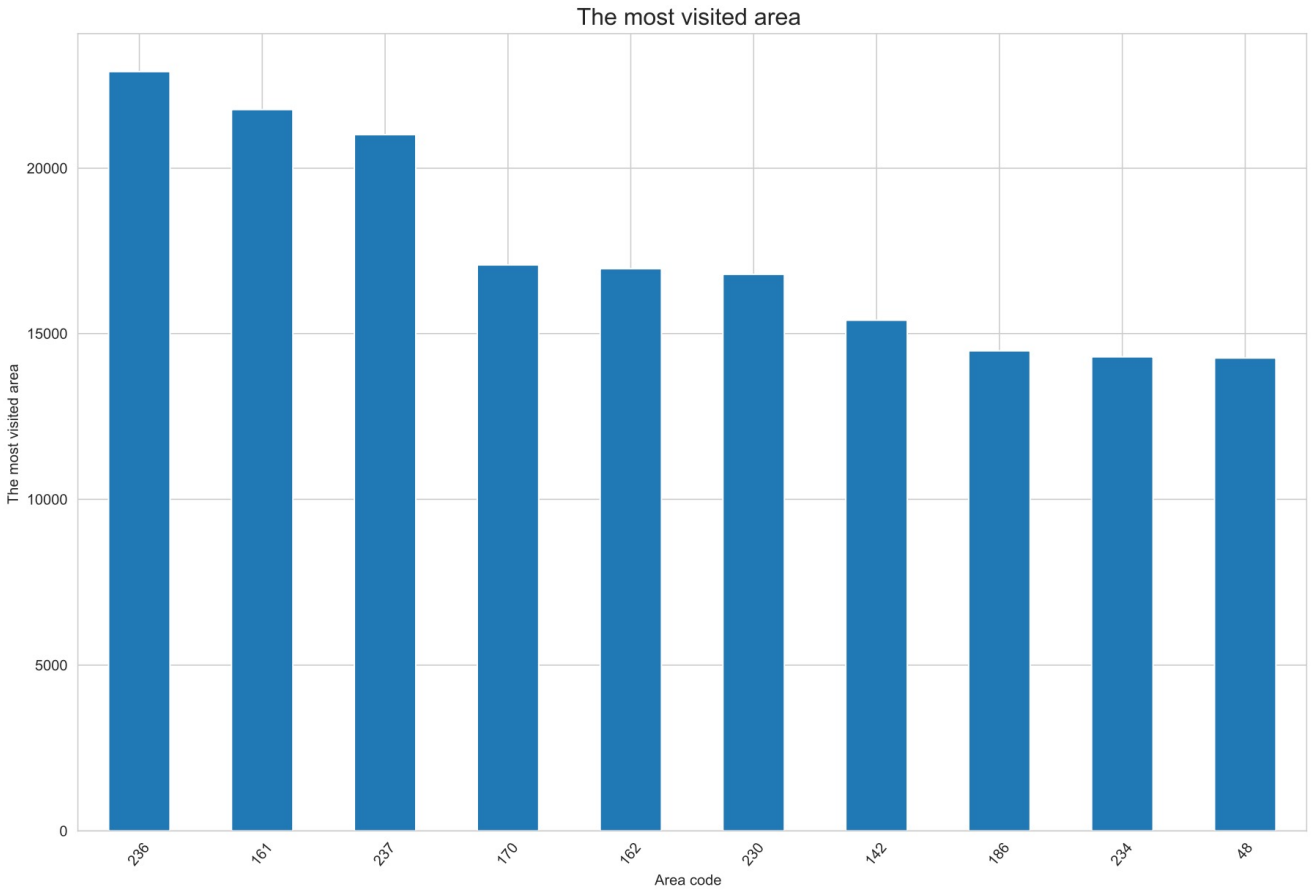
Scatter Plot distance vs Fare amount

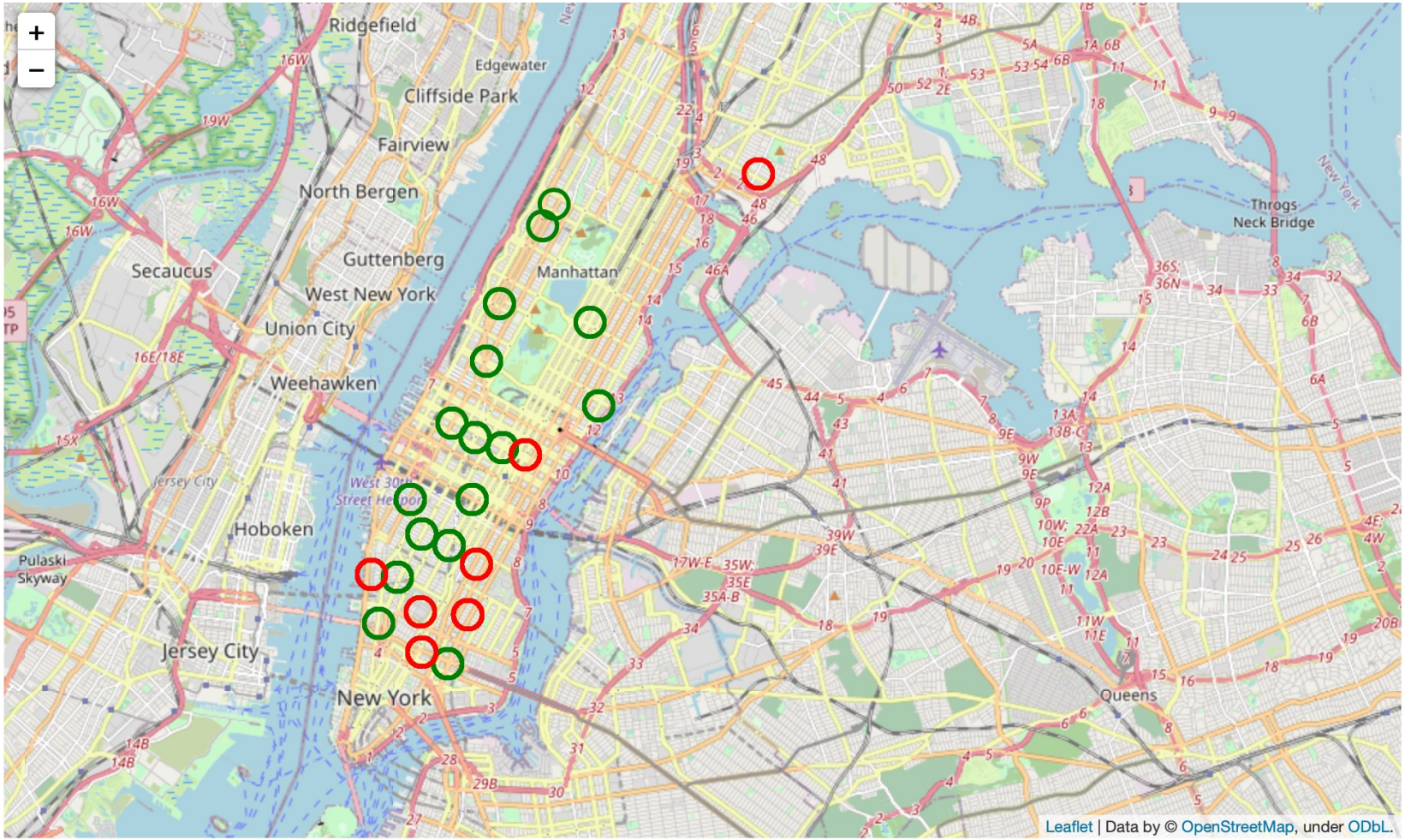


Fare amount for trips in every hour

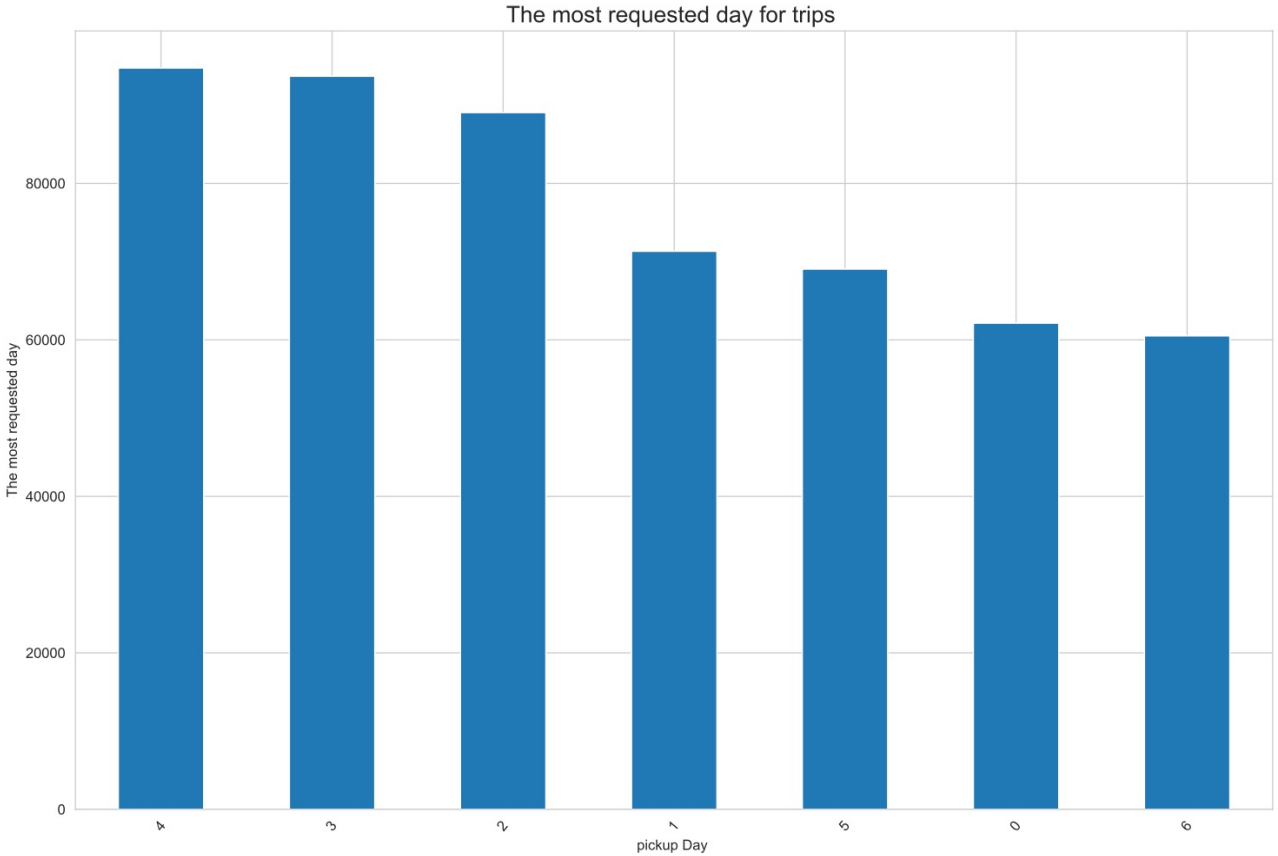


the most visited Area

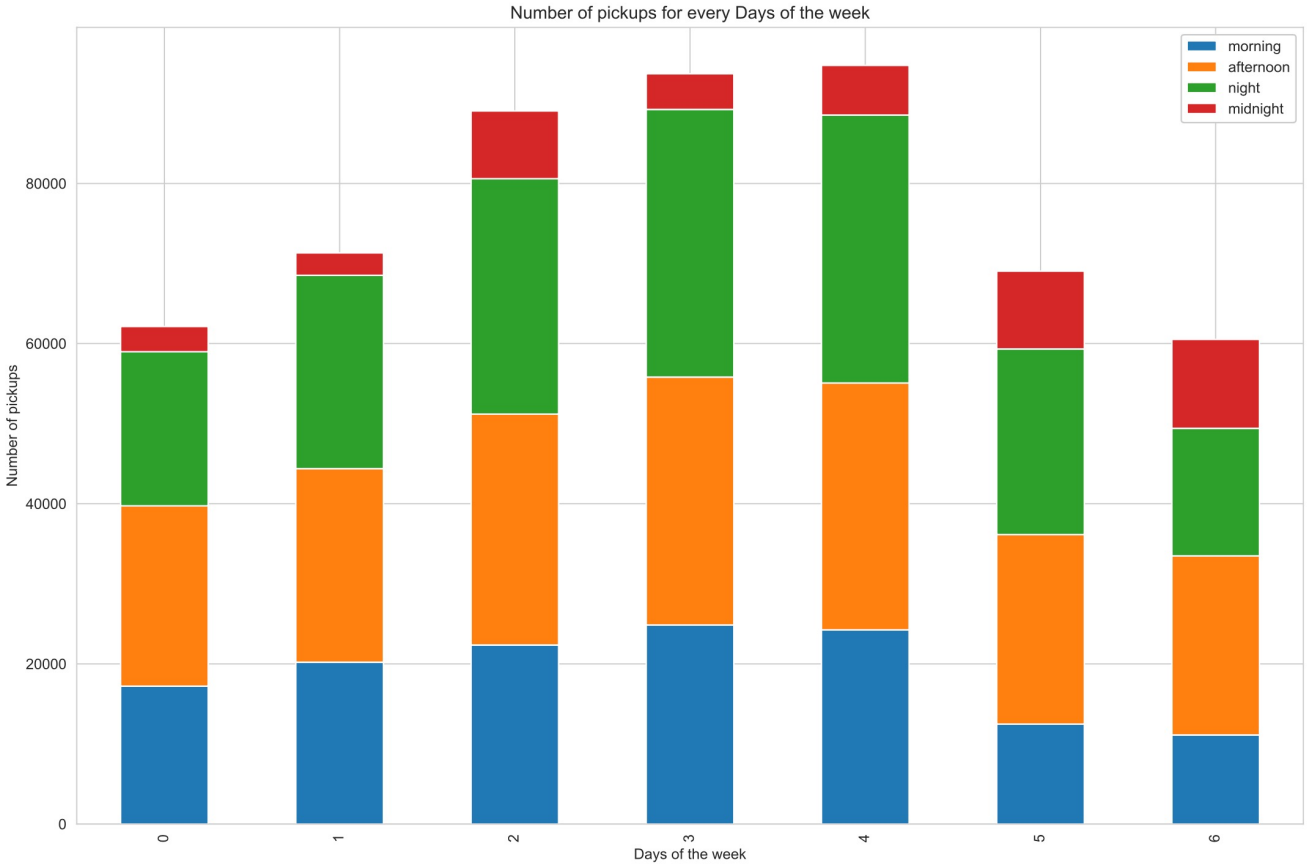



Ridgefield
Edgewater
Cliffside Park
Fairview
North Bergen
Guttenberg
Secaucus
West New York
Union City
Weehawken
Manhattan
Throgs Neck Bridge
Hoboken
Jersey City
Pulaski Skyway
New York
Queens
Leaflet | Data by © OpenStreetMap, under ODbL.

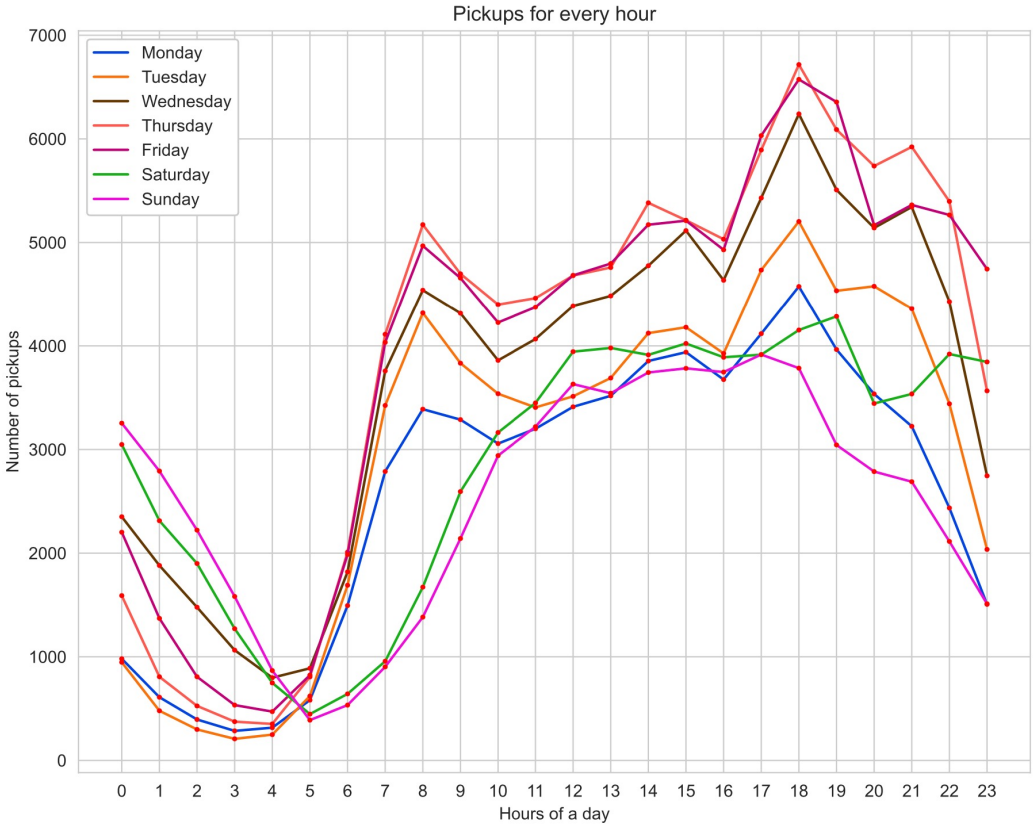
The most requested day for trips



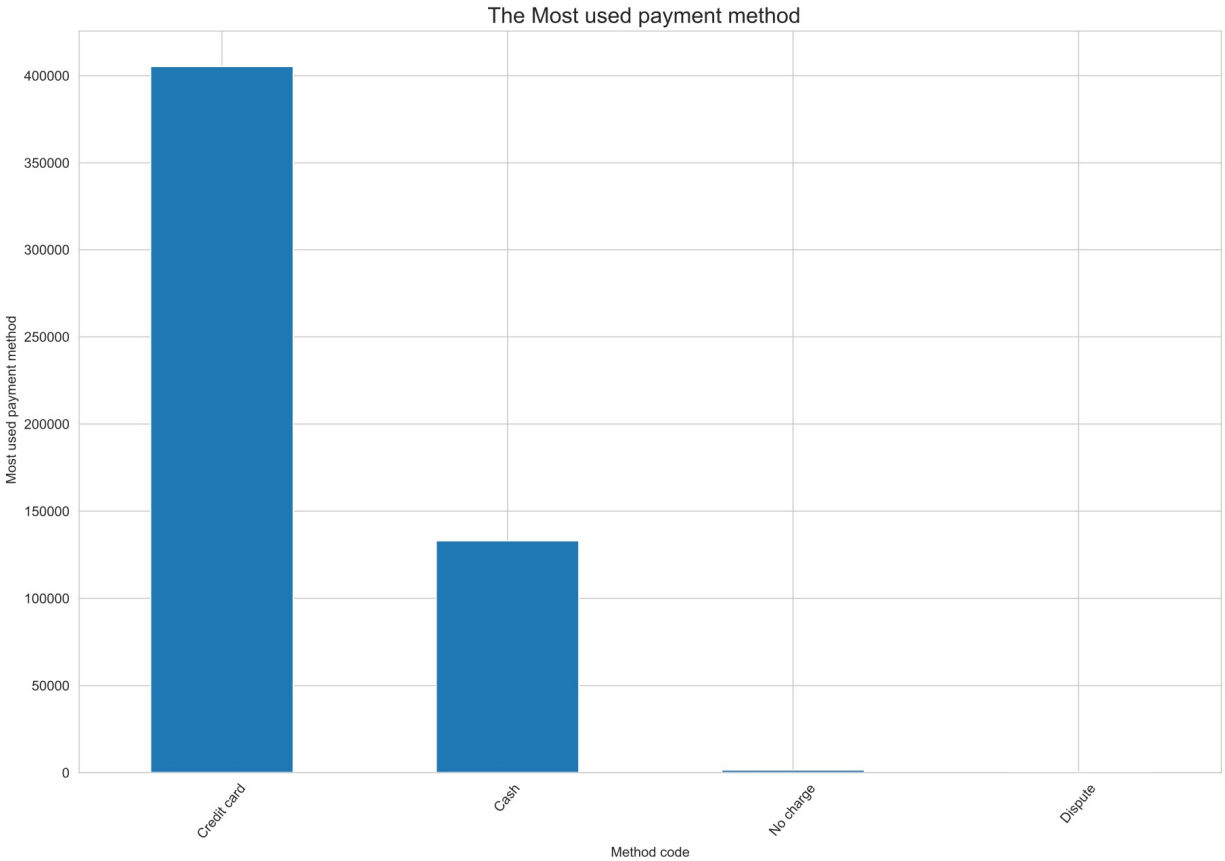
Number of pickups for every Days of the week



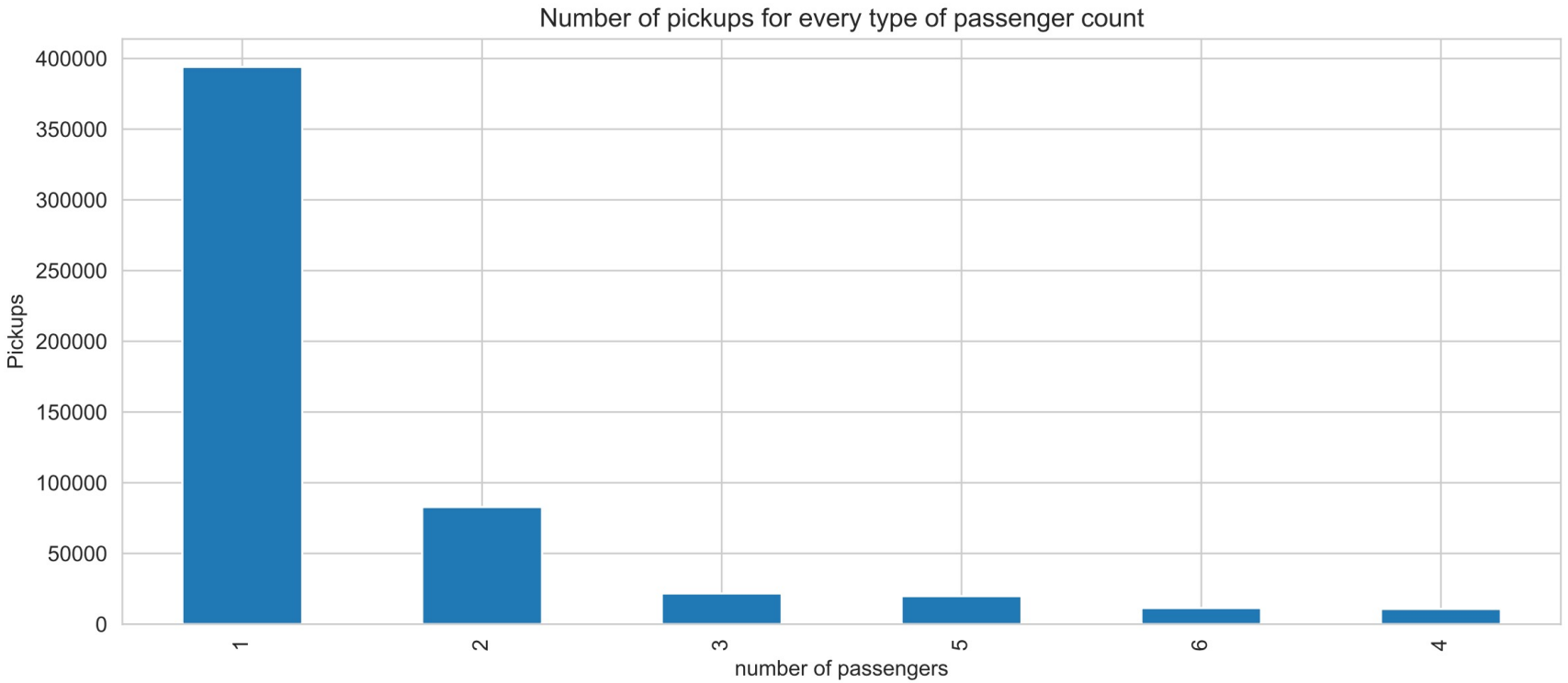
number of trips per hour



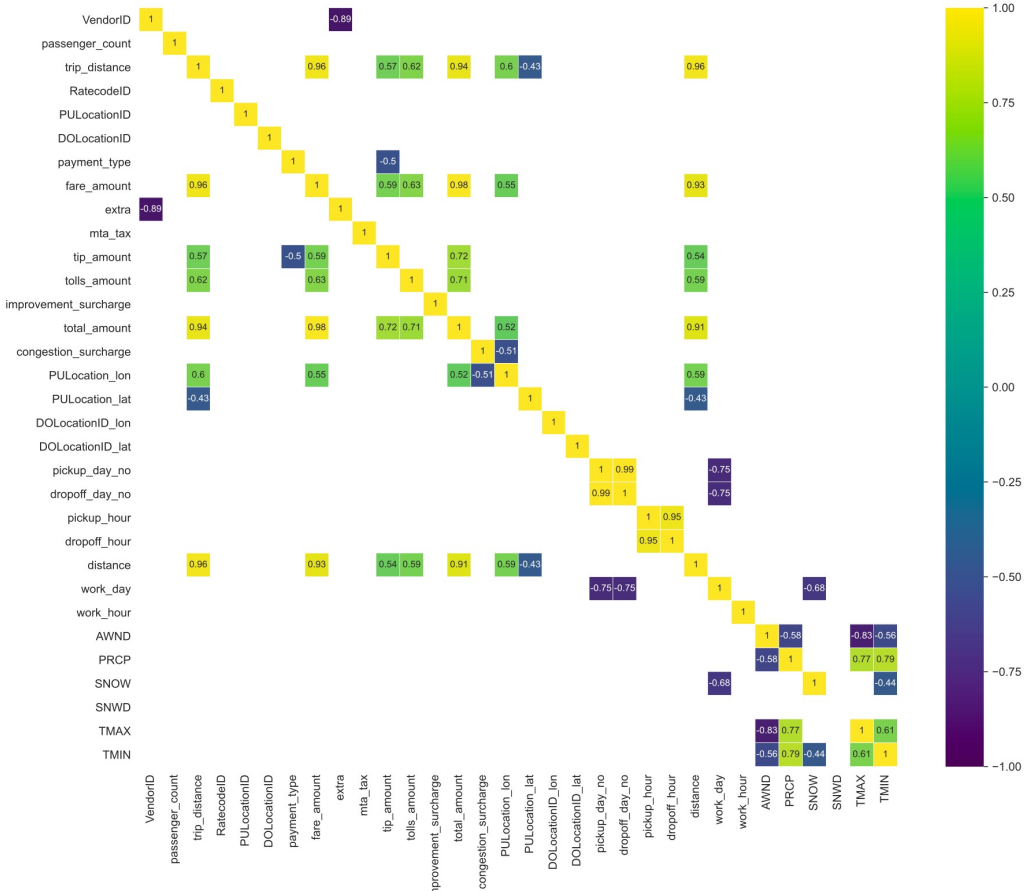
most payment method used

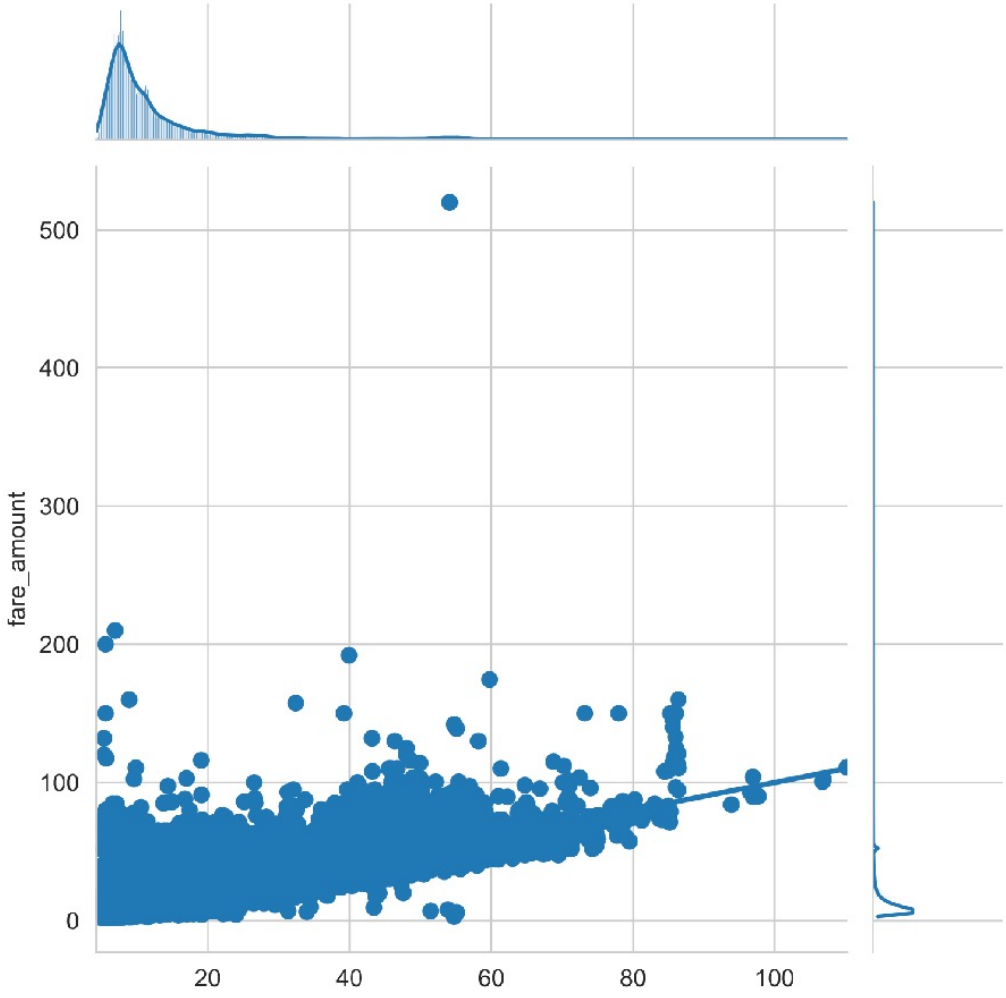


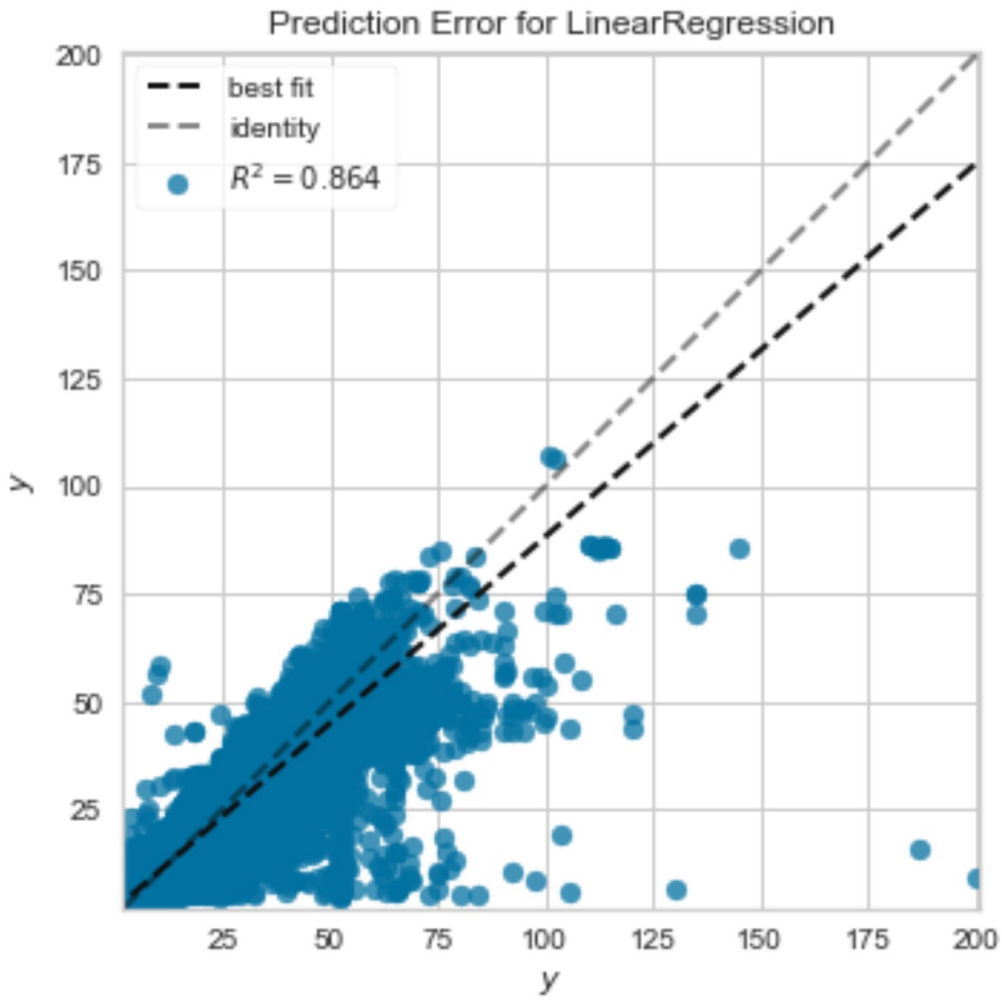
number of pickups for every type of passenger count

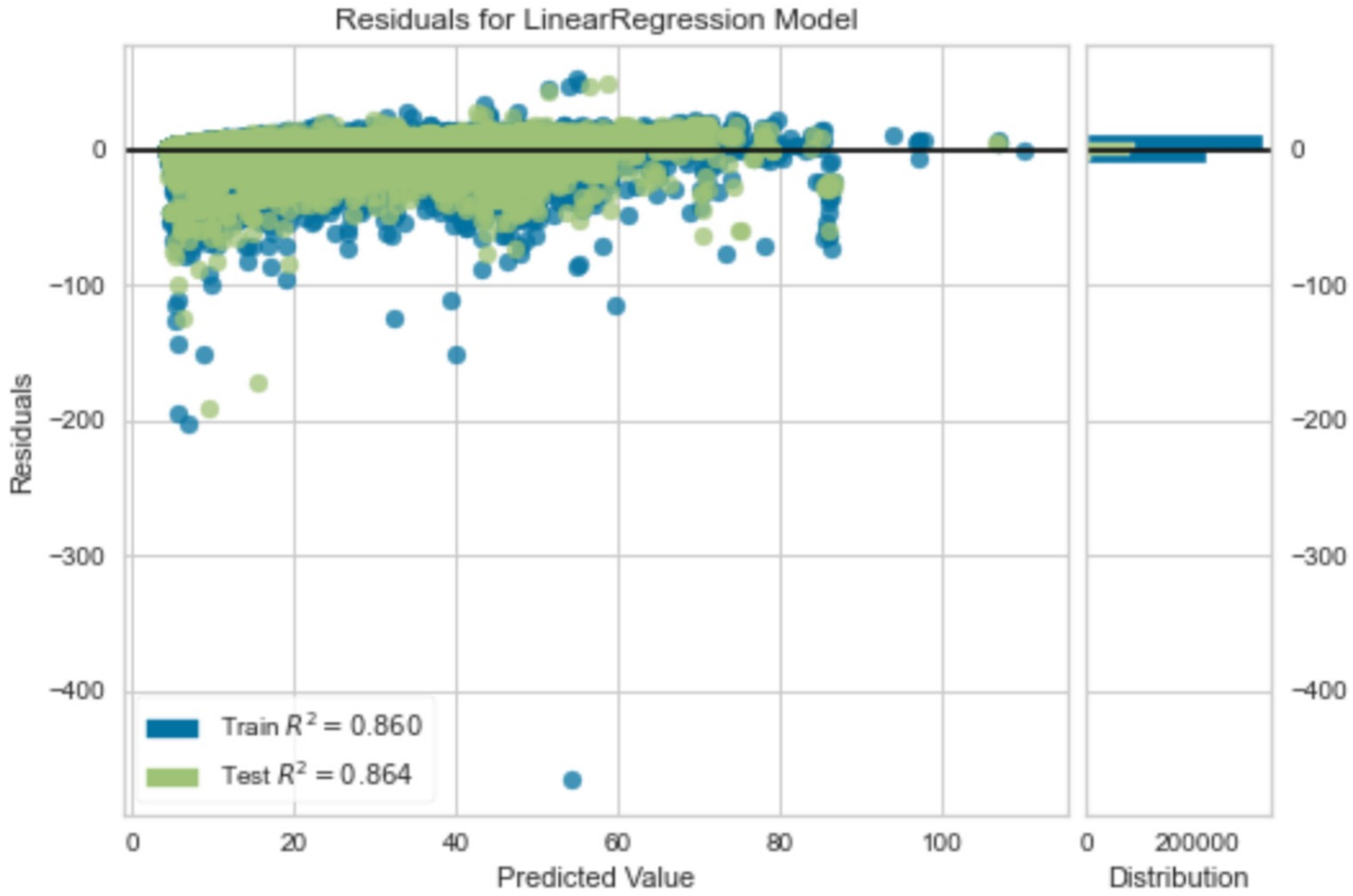


Correlations




fare_amount
500
400
300
200
100
0
20
40
60
80
100


Prediction Error for LinearRegression
best fit
identity
$R^2 = 0.864$
y
y
200
175
150
125
100
75
50
25
25
50
75
100
125
150
175
200


Residuals for LinearRegression Model
Residuals
0
−100
−200
−300
−400
Train $R^2 = 0.860$
Test $R^2 = 0.864$
0
20
40
60
80
100
Predicted Value
0
200000
Distribution

Thanks..