Capstone Project: Improve Efficiency Dispatching Taxis

Table of Contents

Project charter

General information

Project name: Improve Efficiency Dispatching Taxis

Benefits - Business:

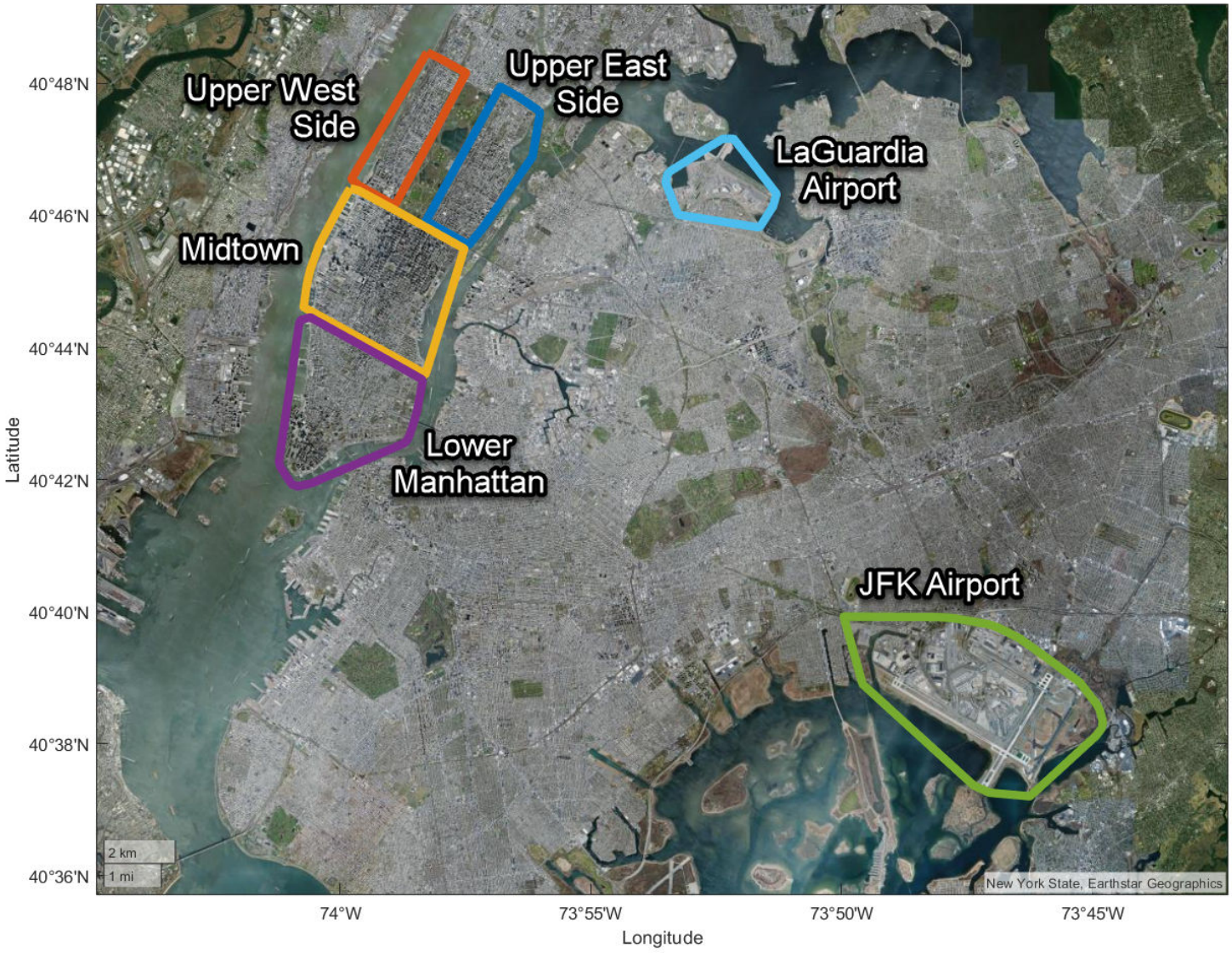
- Increase profitability by dispatching the taxis to the most profitable regions at the most profitable time of the day.
- Reduce waiting time of the taxis.

Benefits - Customer: Reduce waiting time of the customers.

Expected deliverables: Detailed report to take action for optimizing taxi pickup/dropoff locations.

Overview

Problem statement: Mr. Walker, the CEO of Super Taxis, wants to expand the taxi operations to most of Manhattan and airport regions. You are responsible for delivering recommendation on how the company should allocate its taxi fleet on a chosen day.



Goal statement: Predict taxi demand in 2016 at specified areas to dispatch the taxis accordingly, increasing the overall efficiency of the taxi fleet.

Project sponsors: Larry Walker

Project manager: Nguyen Quy Khoi

Deadline: 24/02/2023

Project scope

In-scope:

- Taxi pickup areas: Manhattan (Upper West Side, Upper East Side, Midtown, Lower Manhattan), airports (LaGuardia Airport, JFK Airport).
- Taxi fleet (100 taxi medallions) which costs 50,000,000 USD in total medallion.
- Taxi dispatch data in 2015.

Out-of-scope:

- Guaranteed robustness to unexpected events.
- Comparison to other regions and zones in New York City.

Import and Explore Data

Import and Explore Taxi data

The raw taxi data is included. It involves the following information:

- Number of passengers
- Pickup/dropoff time and location
- Total charge, which is the sum of fare, extra charge, tax, tips, and toll
- TPEP providers: Creative Mobile Technologies, VeriFone Inc.
- Additional data: Taxi pickup/dropoff zones and boroughs. It is included using `addTaxiZones` function.
- Addtional data: Taxi crow distance. It is included using `addCrowDistance` function.

```
taxiData = 2922266×24 table
```

...

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
1	2	2015-01-15 14:...	2015-01-15 14:...	1	3	-73.9643
2	2	2015-01-15 14:...	2015-01-15 14:...	1	0.6700	-73.9709
3	2	2015-01-07 14:...	2015-01-07 15:...	1	0.9800	-73.9487
4	2	2015-01-07 14:...	2015-01-07 15:...	3	4.3900	-73.9887
5	1	2015-01-20 23:...	2015-01-20 23:...	1	3.9000	-73.9750
6	2	2015-01-18 19:...	2015-01-18 20:...	6	4	-73.9710
7	2	2015-01-01 01:...	2015-01-01 01:...	1	5.7800	-74.0078
8	2	2015-01-01 01:...	2015-01-01 01:...	4	0.8800	-73.9642
9	1	2015-01-28 10:...	2015-01-28 10:...	1	0.6000	-73.9664
10	1	2015-01-23 16:...	2015-01-23 17:...	1	9.3000	-74.0067
11	1	2015-01-07 20:...	2015-01-07 20:...	1	6.9000	-73.9901
12	1	2015-01-10 19:...	2015-01-10 19:...	1	1	-73.9785
13	1	2015-01-10 19:...	2015-01-10 19:...	1	1.1000	-74.0016
14	2	2015-01-25 17:...	2015-01-25 17:...	1	0	-73.9757
15	2	2015-01-23 00:...	2015-01-23 00:...	1	6.0300	-73.9852
16	1	2015-01-23 17:...	2015-01-23 18:...	1	8.2000	-73.8744

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
17	2	2015-01-17 19:...	2015-01-17 19:...	1	0.8900	-73.9545
18	2	2015-01-17 19:...	2015-01-17 19:...	1	2.5700	-73.9747
19	1	2015-01-17 23:...	2015-01-17 23:...	3	0.5000	-73.9857
20	1	2015-01-28 20:...	2015-01-28 20:...	1	0.8000	-73.9854
21	1	2015-01-07 21:...	2015-01-07 21:...	1	0.5000	-73.9649
22	2	2015-01-18 22:...	2015-01-18 23:...	1	9.8800	-73.9830
23	1	2015-01-26 14:...	2015-01-26 14:...	1	0.9000	-74.0033
24	2	2015-01-07 19:...	2015-01-07 19:...	2	1.3500	-73.9896
25	2	2015-01-07 19:...	2015-01-07 19:...	1	0.3600	0
26	1	2015-01-15 11:...	2015-01-15 11:...	1	1.2000	-73.9718
27	2	2015-01-05 22:...	2015-01-05 22:...	1	0.6000	-73.9530
28	1	2015-01-15 11:...	2015-01-15 11:...	1	0.4000	-73.9657
29	2	2015-01-30 09:...	2015-01-30 09:...	1	0.2900	-74.0051
30	2	2015-01-27 18:...	2015-01-27 18:...	1	1.6100	-73.9536
31	2	2015-01-27 18:...	2015-01-27 18:...	1	2.0500	-73.9873
32	1	2015-01-21 05:...	2015-01-21 06:...	1	10.4000	-73.9509
33	1	2015-01-07 21:...	2015-01-07 21:...	1	0.9000	-73.9936
34	1	2015-01-18 00:...	2015-01-18 00:...	1	0.7000	-73.9887
35	1	2015-01-23 18:...	2015-01-23 18:...	1	1.8000	-73.9672
36	2	2015-01-28 13:...	2015-01-28 13:...	6	0.4500	-73.9965
37	2	2015-01-25 17:...	2015-01-25 17:...	5	0.5400	-73.9638
38	1	2015-01-10 21:...	2015-01-10 22:...	1	4.5000	-73.9902
39	2	2015-01-18 11:...	2015-01-18 11:...	1	3.0500	-73.9887
40	2	2015-01-22 19:...	2015-01-22 19:...	6	2.5300	-73.9946
41	1	2015-01-15 13:...	2015-01-15 13:...	1	1.3000	-73.9750
42	1	2015-01-23 18:...	2015-01-23 19:...	1	5.8000	-73.9767
43	2	2015-01-10 13:...	2015-01-10 14:...	5	2.9900	-73.9779
44	2	2015-01-10 17:...	2015-01-10 17:...	5	2.0700	-73.9526
45	2	2015-01-10 17:...	2015-01-10 17:...	1	2.2300	-73.9552
46	1	2015-01-28 08:...	2015-01-28 08:...	1	0.4000	-74.0080
47	1	2015-01-21 09:...	2015-01-21 09:...	1	0.7000	-74.0007
48	2	2015-01-30 20:...	2015-01-30 20:...	1	3.1200	-73.9553
49	2	2015-01-13 23:...	2015-01-13 23:...	1	10.8000	-73.8313

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
50	1	2015-01-26 18:...	2015-01-26 19:...	1	2.5000	-73.9859
51	1	2015-01-26 20:...	2015-01-26 20:...	0	1.2000	-73.9875
52	2	2015-01-02 14:...	2015-01-02 14:...	2	1.9900	-73.9779
53	1	2015-01-26 21:...	2015-01-26 21:...	1	1.6000	-73.9823
54	1	2015-01-27 13:...	2015-01-27 13:...	1	3.6000	-73.9621
55	2	2015-01-24 18:...	2015-01-24 18:...	1	0.4400	-73.9699
56	2	2015-01-30 10:...	2015-01-30 10:...	1	1.6500	-73.9618
57	2	2015-01-15 11:...	2015-01-15 11:...	1	0.9600	-73.9631
58	2	2015-01-31 20:...	2015-01-31 20:...	1	0.4500	-73.9847
59	1	2015-01-10 23:...	2015-01-11 00:...	2	6.2000	-73.9618
60	2	2015-01-20 08:...	2015-01-20 09:...	1	3.5600	-73.9923
61	2	2015-01-31 19:...	2015-01-31 19:...	2	4.2100	-73.9773
62	2	2015-01-17 14:...	2015-01-17 14:...	1	0.9300	-73.9801
63	1	2015-01-11 00:...	2015-01-11 01:...	3	1.5000	-73.9247
64	1	2015-01-11 01:...	2015-01-11 01:...	4	0.8000	-73.9801
65	1	2015-01-27 16:...	2015-01-27 16:...	5	1.1000	-73.9730
66	2	2015-01-20 15:...	2015-01-20 15:...	6	0.7800	-73.9719
67	1	2015-01-27 16:...	2015-01-27 16:...	1	3.6000	-73.9711
68	2	2015-01-17 20:...	2015-01-17 20:...	1	2.9400	-73.9882
69	1	2015-01-28 09:...	2015-01-28 09:...	1	3	-73.9600
70	2	2015-01-17 22:...	2015-01-17 22:...	2	3.1800	-73.9948
71	2	2015-01-17 22:...	2015-01-17 22:...	1	1.0600	-73.9810
72	2	2015-01-17 22:...	2015-01-17 22:...	1	0.1100	-74.0024
73	1	2015-01-23 19:...	2015-01-23 19:...	2	1.5000	-73.9872
74	2	2015-01-31 11:...	2015-01-31 11:...	1	6.8900	-73.9794
75	1	2015-01-04 14:...	2015-01-04 14:...	1	1.2000	-73.9634
76	1	2015-01-27 17:...	2015-01-27 17:...	1	1.3000	-73.9813
77	2	2015-01-15 13:...	2015-01-15 13:...	1	1.0600	-73.9885
78	1	2015-01-15 16:...	2015-01-15 17:...	1	17.5000	-73.9735
79	2	2015-01-27 18:...	2015-01-27 19:...	2	3.1500	-73.9845
80	1	2015-01-27 18:...	2015-01-27 18:...	1	1	-73.9881
81	2	2015-01-07 08:...	2015-01-07 09:...	5	2.5400	-73.9989
82	2	2015-01-19 14:...	2015-01-19 14:...	6	0.5600	-73.9789

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
83	1	2015-01-21 08:...	2015-01-21 09:...	1	19.6000	-74.0017
84	2	2015-01-16 20:...	2015-01-16 20:...	3	2.3200	-73.9880
85	2	2015-01-14 12:...	2015-01-14 12:...	1	0.7800	-73.9963
86	2	2015-01-14 12:...	2015-01-14 12:...	1	1.1500	-73.9920
87	1	2015-01-08 00:...	2015-01-08 00:...	1	12.2000	-73.8744
88	1	2015-01-08 00:...	2015-01-08 00:...	1	1.9000	0
89	2	2015-01-09 10:...	2015-01-09 10:...	1	1.0300	-74.0101
90	1	2015-01-04 14:...	2015-01-04 15:...	1	3.6000	-73.9868
91	2	2015-01-06 15:...	2015-01-06 15:...	5	0.8200	-73.9782
92	2	2015-01-27 22:...	2015-01-27 22:...	4	0.6600	-73.9879
93	2	2015-01-07 15:...	2015-01-07 15:...	2	0.4800	-73.9789
94	1	2015-01-15 17:...	2015-01-15 18:...	1	1.3000	-74.0042
95	1	2015-01-15 18:...	2015-01-15 18:...	1	0.8000	-73.9974
96	1	2015-01-27 21:...	2015-01-27 21:...	1	2.6000	-73.9823
97	2	2015-01-28 06:...	2015-01-28 07:...	1	3.3600	-73.9502
98	2	2015-01-24 11:...	2015-01-24 12:...	2	1.8300	-73.9951
99	2	2015-01-03 09:...	2015-01-03 09:...	1	2.7600	-73.9544
100	1	2015-01-03 09:...	2015-01-03 09:...	1	1.6000	-73.9590

⋮

Import and Explore Region data

In the raw taxi data, the New York City is divided into various zones. Depending on one's classification criteria, the zones are grouped into regions. From the project scope, there are 6 regions of interest:

- Manhattan: Upper West Side, Upper East Side, Midtown, Lower Manhattan.
- Airports: LaGuardia Airport, JFK Airport.

The data is imported from the exisiting file `Taxi Regions and Zones.csv`. The classified regions and their zones are shown below:

`TaxiRegionsandZones = 20×6 table`

...

	LowerManhattan	Midtown	UpperEastSide
1	"Lower Manhattan"	"Midtown"	"Upper East Side"
2	"Alphabet City"	"Clinton East"	"Upper East Side North"
3	"Battery Park"	"Clinton West"	"Upper East Side South"
4	"Battery Park City"	"Midtown Center"	"Yorkville East"

	LowerManhattan	Midtown	UpperEastSide
5	"Chinatown"	"Midtown East"	"Yorkville West"
6	"East Village"	"Midtown North"	"Lenox Hill East"
7	"Financial District North"	"Midtown South"	"Lenox Hill West"
8	"Financial District South"	"Murray Hill"	"East Harlem South"
9	"Greenwich Village North"	"Penn Station/Madison Sq West"	""
10	"Greenwich Village South"	"Union Sq"	""
11	"Hudson Sq"	"UN/Turtle Bay South"	""
12	"Little Italy/NoLiTa"	"Times Sq/Theatre District"	""
13	"Lower East Side"	"Sutton Place/Turtle Bay North"	""
14	"Meatpacking/West Village ...	"Stuy Town/Peter Cooper Village"	""
15	"Seaport"	"West Chelsea/Hudson Yards"	""
16	"SoHo"	"East Chelsea"	""
17	"TriBeCa/Civic Center"	"Garment District"	""
18	"Two Bridges/Seward Park"	"Gramercy"	""
19	"West Village"	"Flatiron"	""
20	"World Trade Center"	"Kips Bay"	""

Preprocess data: Taxi pickups

Data cleaning

Restructuring

The `TaxiRegionsandZones` table has missing and duplicate cells that needs removing upon inspection. The table must be stacked to perform **data restructuring**, i.e. joining table `TaxiRegionsandZones` and `taxiData`. The joining variable is the New York City's zones. Both pickup and dropoff zones are considered to gain insights from the raw data by grouping.

`TaxiRegionsandZones = 57×2 table`

	Region	Zone
1	LowerManhattan	Alphabet City
2	LowerManhattan	Battery Park
3	LowerManhattan	Battery Park City
4	LowerManhattan	Chinatown
5	LowerManhattan	East Village
6	LowerManhattan	Financial District North
7	LowerManhattan	Financial District South
8	LowerManhattan	Greenwich Village North

	Region	Zone
9	LowerManhattan	Greenwich Village South
10	LowerManhattan	Hudson Sq
11	LowerManhattan	Little Italy/NoLiTa
12	LowerManhattan	Lower East Side
13	LowerManhattan	Lower Manhattan
14	LowerManhattan	Meatpacking/West Village...
15	LowerManhattan	Seaport
16	LowerManhattan	SoHo
17	LowerManhattan	TriBeCa/Civic Center
18	LowerManhattan	Two Bridges/Seward Park
19	LowerManhattan	West Village
20	LowerManhattan	World Trade Center
21	Midtown	Clinton East
22	Midtown	Clinton West
23	Midtown	East Chelsea
24	Midtown	Flatiron
25	Midtown	Garment District
26	Midtown	Gramercy
27	Midtown	Kips Bay
28	Midtown	Midtown
29	Midtown	Midtown Center
30	Midtown	Midtown East
31	Midtown	Midtown North
32	Midtown	Midtown South
33	Midtown	Murray Hill
34	Midtown	Penn Station/Madison Sq ...
35	Midtown	Stuy Town/Peter Cooper V...
36	Midtown	Sutton Place/Turtle Bay ...
37	Midtown	Times Sq/Theatre District
38	Midtown	UN/Turtle Bay South
39	Midtown	Union Sq
40	Midtown	West Chelsea/Hudson Yards
41	UpperEastSide	East Harlem South

	Region	Zone
42	UpperEastSide	Lenox Hill East
43	UpperEastSide	Lenox Hill West
44	UpperEastSide	Upper East Side
45	UpperEastSide	Upper East Side North
46	UpperEastSide	Upper East Side South
47	UpperEastSide	Yorkville East
48	UpperEastSide	Yorkville West
49	UpperWestSide	Bloomingdale
50	UpperWestSide	Lincoln Square East
51	UpperWestSide	Lincoln Square West
52	UpperWestSide	Manhattan Valley
53	UpperWestSide	Upper West Side
54	UpperWestSide	Upper West Side North
55	UpperWestSide	Upper West Side South
56	JFKAirport	JFK Airport
57	LaGuardiaAirp...	LaGuardia Airport

Remove

The data is cleaned as follows:

- Apply cleaning operations using the function `basicPreprocessing.mlx`, excluding the applied functions for a typical trip and typical charges.
- Exclude trips with durations below 1 minute and over 3 hours.
- Exclude any trips above New York's speed limit. According to Speed-Limits.com, the highest speed limit is 55 mph, excluding rural freeways.
- Exclude any trips outside of New York City. According to Walks of New York, the longest crow distance between New York's boundaries is about 35 miles. Since a taxi must go through existing streets, the longest trip is assumed to be 70 miles, which is doubled the boundary distance.
- Exclude any trips with tolls outside the 1st and 99th percentile range.
- Exclude any trips with fares below 2.5 USD and above 75 USD. According to The Hill, the base fare in 2015 was 2.5 USD.
- Exclude any trips with a total charge below 2.5 USD and above 100 USD. According to The Hill, the base fare in 2015 was 2.5 USD.

```
taxiDataCleaned = 2711379×30 table
```

...

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
1	2	2015-08-21 09:...	2015-08-21 09:...	5	1.8300	-73.8419
2	2	2015-08-21 10:...	2015-08-21 10:...	5	1.1400	-73.8419
3	2	2015-10-03 08:...	2015-10-03 09:...	5	2.8000	-73.8420
4	1	2015-06-07 03:...	2015-06-07 03:...	2	14.1000	-73.9742
5	1	2015-04-03 09:...	2015-04-03 10:...	1	11.1000	-73.9102
6	1	2015-11-09 20:...	2015-11-09 21:...	1	17.6000	-74.0144
7	2	2015-03-03 17:...	2015-03-03 17:...	1	0.6500	-73.8638
8	2	2015-04-03 00:...	2015-04-03 00:...	1	10.6000	-73.9454
9	1	2015-04-07 20:...	2015-04-07 20:...	1	9.1000	-73.9514
10	2	2015-02-22 20:...	2015-02-22 20:...	5	10.1400	-73.9429
11	1	2015-11-10 00:...	2015-11-10 01:...	1	16.1000	-73.9976
12	2	2015-01-12 10:...	2015-01-12 10:...	1	14.3800	-73.9899
13	1	2015-03-21 23:...	2015-03-22 00:...	2	14.8000	-73.9865
14	1	2015-08-29 22:...	2015-08-29 22:...	1	16.6000	-73.9948
15	2	2015-11-16 22:...	2015-11-16 22:...	5	13.0700	-73.9884
16	1	2015-12-16 22:...	2015-12-16 23:...	1	14	-73.9890
17	1	2015-04-03 01:...	2015-04-03 01:...	1	6.7000	-73.8964
18	2	2015-09-24 20:...	2015-09-24 20:...	2	19.7400	-73.9895
19	1	2015-11-07 07:...	2015-11-07 08:...	1	14.9000	-73.9980
20	1	2015-02-10 18:...	2015-02-10 19:...	2	7.6000	-73.9367
21	2	2015-07-26 22:...	2015-07-26 22:...	1	8.2000	-73.9381
22	2	2015-11-13 00:...	2015-11-13 01:...	1	9.4600	-73.9333
23	2	2015-12-21 22:...	2015-12-21 22:...	5	8.5500	-73.9412
24	2	2015-04-06 10:...	2015-04-06 11:...	6	11.1300	-73.9527
25	2	2015-06-14 02:...	2015-06-14 03:...	1	16.0900	-73.9893
26	1	2015-09-17 07:...	2015-09-17 08:...	1	15.6000	-73.9900
27	2	2015-04-11 00:...	2015-04-11 00:...	1	0.2600	-73.8461
28	1	2015-03-12 22:...	2015-03-12 22:...	1	16.5000	-74.0057
29	1	2015-03-20 00:...	2015-03-20 01:...	1	16	-73.9983
30	2	2015-06-29 01:...	2015-06-29 02:...	2	15.5800	-73.9941
31	2	2015-12-19 09:...	2015-12-19 09:...	1	15.8600	-73.9931
32	1	2015-04-27 20:...	2015-04-27 21:...	1	15.1000	-73.9877
33	2	2015-04-19 21:...	2015-04-19 22:...	1	13.6600	-73.9900

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
34	2	2015-08-05 22:...	2015-08-05 22:...	1	14.8700	-73.9879
35	1	2015-11-14 01:...	2015-11-14 02:...	1	14.5000	-73.9897
36	1	2015-01-15 03:...	2015-01-15 04:...	2	13.9000	-73.9854
37	2	2015-01-17 19:...	2015-01-17 19:...	5	14.2200	-73.9819
38	2	2015-06-19 02:...	2015-06-19 02:...	5	13.5900	-73.9880
39	1	2015-12-06 19:...	2015-12-06 20:...	2	15.7000	-73.9898
40	2	2015-04-10 22:...	2015-04-10 22:...	1	15.7400	-73.9905
41	2	2015-11-29 02:...	2015-11-29 03:...	1	16.3800	-73.9912
42	1	2015-05-24 03:...	2015-05-24 04:...	1	17.9000	-73.9999
43	2	2015-07-08 02:...	2015-07-08 02:...	3	8.7000	-73.9209
44	2	2015-04-10 01:...	2015-04-10 02:...	3	7.6100	-73.9309
45	2	2015-03-18 20:...	2015-03-18 21:...	1	18.3100	-74.0071
46	2	2015-06-27 07:...	2015-06-27 08:...	5	19.4500	-74.0090
47	1	2015-09-04 00:...	2015-09-04 00:...	1	4.2000	-73.9204
48	2	2015-01-25 00:...	2015-01-25 00:...	5	17.8100	-73.7894
49	1	2015-02-22 15:...	2015-02-22 15:...	2	17.7000	-73.7882
50	2	2015-02-22 15:...	2015-02-22 15:...	1	18.8900	-73.7826
51	1	2015-02-16 20:...	2015-02-16 21:...	2	18.4000	-73.7826
52	1	2015-03-14 16:...	2015-03-14 16:...	1	17.7000	-73.7821
53	2	2015-01-17 00:...	2015-01-17 00:...	1	10.6700	-73.8764
54	2	2015-01-13 09:...	2015-01-13 09:...	1	13.7800	-73.8658
55	1	2015-02-03 16:...	2015-02-03 17:...	1	11.8000	-73.8745
56	2	2015-02-26 13:...	2015-02-26 14:...	1	11.6800	-73.8628
57	2	2015-03-10 14:...	2015-03-10 14:...	3	10.8400	-73.8712
58	2	2015-06-11 23:...	2015-06-12 00:...	2	14.3300	-73.9543
59	2	2015-08-15 08:...	2015-08-15 08:...	1	12.1200	-73.9548
60	2	2015-12-25 16:...	2015-12-25 17:...	1	12.9300	-73.9592
61	2	2015-08-20 18:...	2015-08-20 19:...	6	11.2400	-73.9653
62	2	2015-04-08 22:...	2015-04-08 22:...	1	12.8900	-73.9819
63	2	2015-08-21 01:...	2015-08-21 02:...	1	15.3600	-73.9947
64	1	2015-03-29 04:...	2015-03-29 04:...	2	15.3000	-73.9474
65	1	2015-10-08 00:...	2015-10-08 01:...	2	16.4000	-73.9936
66	1	2015-03-15 13:...	2015-03-15 13:...	1	9.8000	-73.9544

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
67	1	2015-01-28 14:...	2015-01-28 15:...	2	12.3000	-73.9769
68	1	2015-05-12 22:...	2015-05-12 23:...	2	12.5000	-73.9783
69	2	2015-06-18 06:...	2015-06-18 06:...	5	13.6400	-73.9815
70	1	2015-08-07 21:...	2015-08-07 21:...	3	12.4000	-73.9781
71	2	2015-09-25 11:...	2015-09-25 12:...	1	14.4500	-73.9787
72	2	2015-12-19 16:...	2015-12-19 17:...	5	13.1200	-73.9690
73	2	2015-11-04 02:...	2015-11-04 03:...	1	14.3600	-73.9816
74	2	2015-01-11 04:...	2015-01-11 05:...	1	13.0500	-73.9866
75	1	2015-01-18 05:...	2015-01-18 05:...	1	15.1000	-73.9886
76	1	2015-02-21 00:...	2015-02-21 00:...	1	14.5000	-73.9843
77	2	2015-11-08 03:...	2015-11-08 03:...	6	15.1900	-73.9879
78	1	2015-11-21 03:...	2015-11-21 04:...	1	15.8000	-73.9871
79	2	2015-06-24 03:...	2015-06-24 03:...	1	10.2500	-73.9666
80	2	2015-08-08 02:...	2015-08-08 02:...	1	10.3100	-73.9636
81	2	2015-07-21 15:...	2015-07-21 16:...	1	9.0400	-73.9059
82	1	2015-04-29 22:...	2015-04-29 22:...	1	7.6000	-73.9220
83	2	2015-02-06 23:...	2015-02-07 00:...	1	15.2000	-73.9776
84	1	2015-09-13 02:...	2015-09-13 03:...	1	20.9000	-73.9785
85	1	2015-11-06 21:...	2015-11-06 22:...	1	12.3000	-73.9767
86	1	2015-12-18 01:...	2015-12-18 02:...	1	13.3000	-73.9742
87	2	2015-02-13 20:...	2015-02-13 20:...	1	1.0700	-73.8710
88	1	2015-07-11 21:...	2015-07-11 21:...	2	3.3000	-73.8616
89	1	2015-07-31 07:...	2015-07-31 07:...	1	1.8000	-73.8517
90	1	2015-02-28 20:...	2015-02-28 21:...	1	15.5000	-73.9941
91	1	2015-02-15 09:...	2015-02-15 10:...	2	17.7000	-73.9909
92	2	2015-03-16 20:...	2015-03-16 21:...	1	18.3200	-73.9937
93	2	2015-08-21 14:...	2015-08-21 15:...	5	13.9400	-73.9943
94	2	2015-09-05 12:...	2015-09-05 12:...	1	15.4100	-73.9931
95	1	2015-10-04 20:...	2015-10-04 20:...	1	14.8000	-73.9940
96	1	2015-11-01 23:...	2015-11-02 00:...	1	13.6000	-73.9932
97	1	2015-02-23 01:...	2015-02-23 01:...	1	4.6000	-73.8588
98	1	2015-12-27 04:...	2015-12-27 04:...	1	5.7000	-73.8585
99	2	2015-02-26 12:...	2015-02-26 13:...	1	16.5700	-73.9828

	Vendor	PickupTime	DropoffTime	Passengers	Distance	PickupLon
100	2	2015-07-31 21:...	2015-07-31 22:...	1	12.8900	-73.9793

⋮

After cleaning, the data is grouped such that:

- All pickups and dropoffs are binned in 1 hour.
- Day of the year is extracted from the binned time.
- Hour in a day is extracted from the binned time.
- Distance, duration, and total charge in 1 hour per region are calculated in mean, sum, and median values.

The summarized table also introduces net pickup variable, which equals to the difference between the number of taxi pickups and dropoffs in 1 hour. If the net pickups are:

- more than 15, the region is in **high** demand.
- between 0 and 15, the region is in **medium** demand.
- smaller than 0, the region is in **low** demand.

In addition, 2 time features are added for training the model: Hour of the day, and day of the year.

`taxiSummary = 43294×17 table`

...

	Time	Day	Hour	Region	PickupCount	DropoffCount
1	2015-01-01 00:...	1	0	LowerManhattan	56	41
2	2015-01-01 00:...	1	0	Midtown	118	72
3	2015-01-01 00:...	1	0	UpperEastSide	39	29
4	2015-01-01 00:...	1	0	UpperWestSide	23	16
5	2015-01-01 01:...	1	1	LowerManhattan	57	51
6	2015-01-01 01:...	1	1	Midtown	102	85
7	2015-01-01 01:...	1	1	UpperEastSide	40	40
8	2015-01-01 01:...	1	1	UpperWestSide	37	31
9	2015-01-01 02:...	1	2	LowerManhattan	57	51
10	2015-01-01 02:...	1	2	Midtown	101	95
11	2015-01-01 02:...	1	2	UpperEastSide	33	48
12	2015-01-01 02:...	1	2	UpperWestSide	23	16
13	2015-01-01 03:...	1	3	LowerManhattan	47	40
14	2015-01-01 03:...	1	3	Midtown	82	67
15	2015-01-01 03:...	1	3	UpperEastSide	21	34
16	2015-01-01 03:...	1	3	UpperWestSide	15	15

	Time	Day	Hour	Region	PickupCount	DropoffCount
17	2015-01-01 04:...	1	4	LowerManhattan	44	26
18	2015-01-01 04:...	1	4	Midtown	66	41
19	2015-01-01 04:...	1	4	UpperEastSide	15	18
20	2015-01-01 04:...	1	4	UpperWestSide	6	4
21	2015-01-01 04:...	1	4	LaGuardiaAirp...	1	2
22	2015-01-01 05:...	1	5	LowerManhattan	18	14
23	2015-01-01 05:...	1	5	Midtown	39	31
24	2015-01-01 05:...	1	5	UpperEastSide	3	8
25	2015-01-01 05:...	1	5	UpperWestSide	6	6
26	2015-01-01 06:...	1	6	LowerManhattan	19	9
27	2015-01-01 06:...	1	6	Midtown	30	18
28	2015-01-01 06:...	1	6	UpperEastSide	6	12
29	2015-01-01 06:...	1	6	UpperWestSide	5	1
30	2015-01-01 06:...	1	6	JFKAirport	2	1
31	2015-01-01 07:...	1	7	LowerManhattan	14	7
32	2015-01-01 07:...	1	7	Midtown	27	28
33	2015-01-01 07:...	1	7	UpperEastSide	8	9
34	2015-01-01 07:...	1	7	UpperWestSide	3	3
35	2015-01-01 07:...	1	7	JFKAirport	2	2
36	2015-01-01 08:...	1	8	LowerManhattan	10	10
37	2015-01-01 08:...	1	8	Midtown	25	13
38	2015-01-01 08:...	1	8	UpperEastSide	7	4
39	2015-01-01 08:...	1	8	UpperWestSide	1	2
40	2015-01-01 08:...	1	8	LaGuardiaAirp...	1	2
41	2015-01-01 09:...	1	9	LowerManhattan	14	9
42	2015-01-01 09:...	1	9	Midtown	43	33
43	2015-01-01 09:...	1	9	UpperEastSide	9	8
44	2015-01-01 09:...	1	9	UpperWestSide	7	10
45	2015-01-01 09:...	1	9	JFKAirport	1	3
46	2015-01-01 09:...	1	9	LaGuardiaAirp...	3	2
47	2015-01-01 10:...	1	10	LowerManhattan	22	17
48	2015-01-01 10:...	1	10	Midtown	50	53
49	2015-01-01 10:...	1	10	UpperEastSide	21	8

	Time	Day	Hour	Region	PickupCount	DropoffCount
50	2015-01-01 10:...	1	10	UpperWestSide	9	7
51	2015-01-01 10:...	1	10	LaGuardiaAirp...	4	1
52	2015-01-01 11:...	1	11	LowerManhattan	28	25
53	2015-01-01 11:...	1	11	Midtown	68	63
54	2015-01-01 11:...	1	11	UpperEastSide	27	23
55	2015-01-01 11:...	1	11	UpperWestSide	14	14
56	2015-01-01 11:...	1	11	JFKAirport	2	5
57	2015-01-01 11:...	1	11	LaGuardiaAirp...	3	2
58	2015-01-01 12:...	1	12	LowerManhattan	37	35
59	2015-01-01 12:...	1	12	Midtown	80	69
60	2015-01-01 12:...	1	12	UpperEastSide	28	24
61	2015-01-01 12:...	1	12	UpperWestSide	19	16
62	2015-01-01 12:...	1	12	JFKAirport	4	1
63	2015-01-01 12:...	1	12	LaGuardiaAirp...	3	2
64	2015-01-01 13:...	1	13	LowerManhattan	45	40
65	2015-01-01 13:...	1	13	Midtown	97	98
66	2015-01-01 13:...	1	13	UpperEastSide	22	19
67	2015-01-01 13:...	1	13	UpperWestSide	16	16
68	2015-01-01 13:...	1	13	JFKAirport	4	3
69	2015-01-01 13:...	1	13	LaGuardiaAirp...	8	6
70	2015-01-01 14:...	1	14	LowerManhattan	35	37
71	2015-01-01 14:...	1	14	Midtown	77	82
72	2015-01-01 14:...	1	14	UpperEastSide	31	27
73	2015-01-01 14:...	1	14	UpperWestSide	18	14
74	2015-01-01 14:...	1	14	LaGuardiaAirp...	3	6
75	2015-01-01 15:...	1	15	LowerManhattan	52	50
76	2015-01-01 15:...	1	15	Midtown	87	75
77	2015-01-01 15:...	1	15	UpperEastSide	33	32
78	2015-01-01 15:...	1	15	UpperWestSide	21	22
79	2015-01-01 15:...	1	15	JFKAirport	3	2
80	2015-01-01 15:...	1	15	LaGuardiaAirp...	3	2
81	2015-01-01 16:...	1	16	LowerManhattan	32	29
82	2015-01-01 16:...	1	16	Midtown	72	86

	Time	Day	Hour	Region	PickupCount	DropoffCount
83	2015-01-01 16:...	1	16	UpperEastSide	30	25
84	2015-01-01 16:...	1	16	UpperWestSide	13	16
85	2015-01-01 16:...	1	16	JFKAirport	5	1
86	2015-01-01 17:...	1	17	LowerManhattan	38	32
87	2015-01-01 17:...	1	17	Midtown	92	92
88	2015-01-01 17:...	1	17	UpperEastSide	32	34
89	2015-01-01 17:...	1	17	UpperWestSide	30	22
90	2015-01-01 17:...	1	17	JFKAirport	5	4
91	2015-01-01 17:...	1	17	LaGuardiaAirp...	7	3
92	2015-01-01 18:...	1	18	LowerManhattan	38	35
93	2015-01-01 18:...	1	18	Midtown	84	92
94	2015-01-01 18:...	1	18	UpperEastSide	27	30
95	2015-01-01 18:...	1	18	UpperWestSide	14	13
96	2015-01-01 18:...	1	18	JFKAirport	3	1
97	2015-01-01 19:...	1	19	LowerManhattan	30	35
98	2015-01-01 19:...	1	19	Midtown	79	71
99	2015-01-01 19:...	1	19	UpperEastSide	25	28
100	2015-01-01 19:...	1	19	UpperWestSide	13	11

⋮

Feature extraction

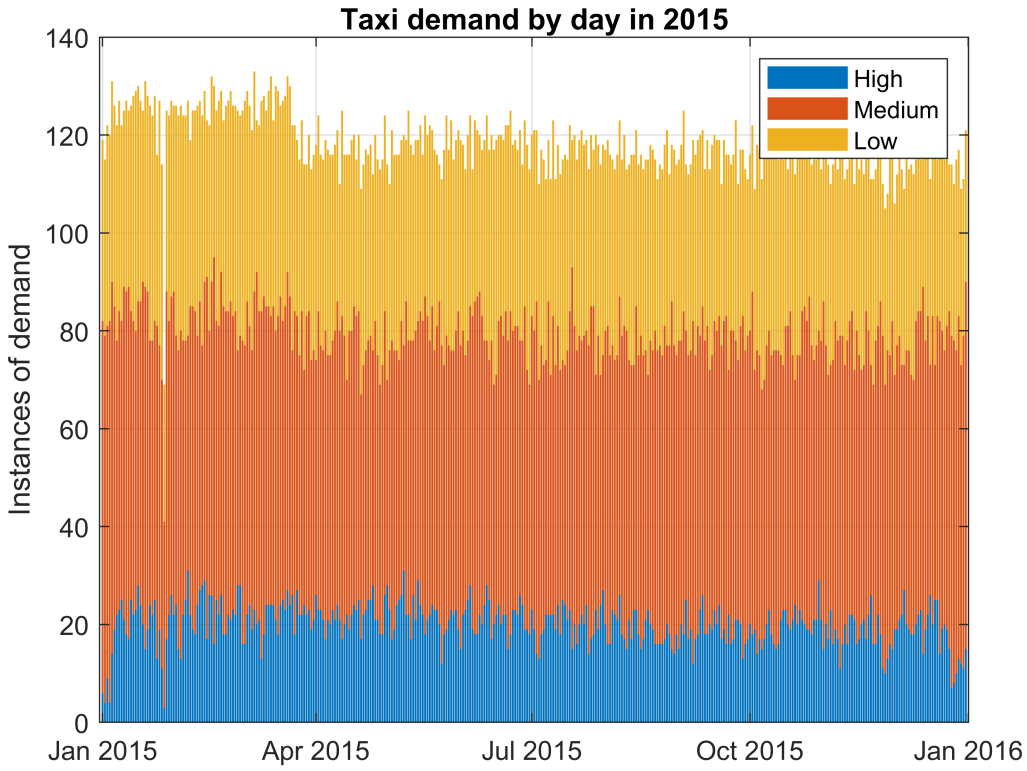
This section is the most time consuming since it requires specific knowledge regarding path optimization for taxis. Because the author has limited experience in this area, only very simple interpretations are shown. In summary, the feature extraction process for this capstone project has 5 steps:

1. Validate the relationship between demand and total charge. The reason is to assess the effectiveness of the predictive model in terms of revenue. If the model predicts demand correctly, it is safe to conclude the revenue will be more likely to increase.
2. Validate the relationship between demand and bank holiday. The reason is to include another predictor for the machine learning model.

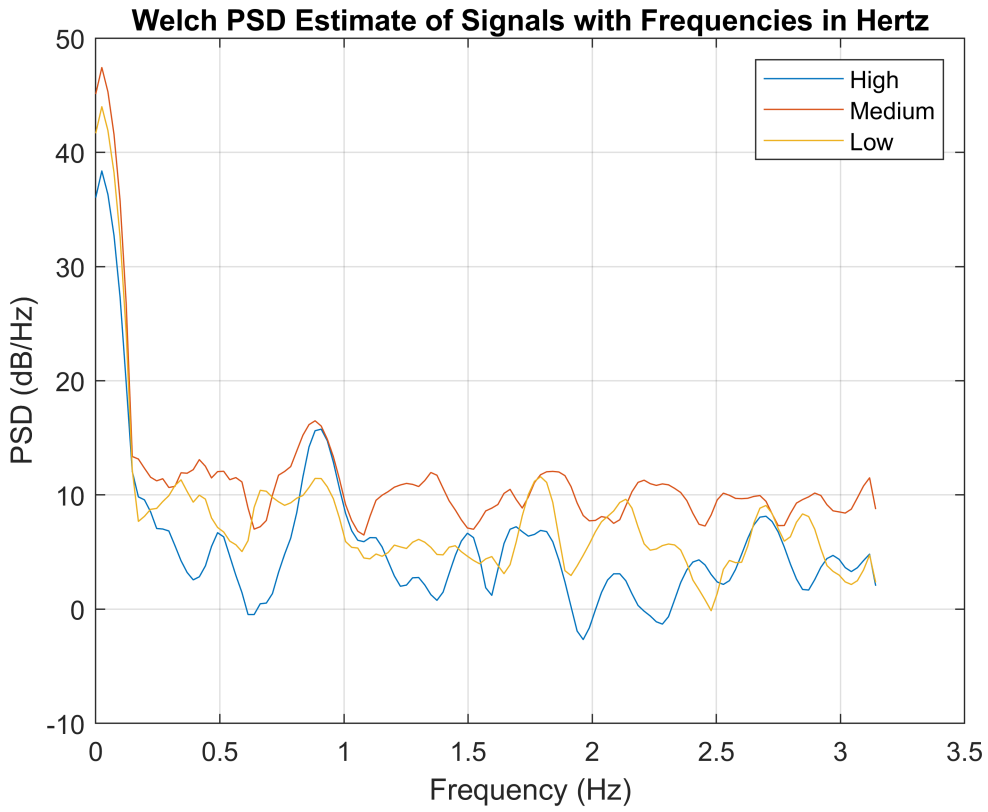
At first glance, the demand for taxi in 2015 can be summarized in the table below. The result shows the imbalance between the classes are not significant enough for under/oversampling (the smallest class 10 times less than the highest class).

Value	Count	Percent
High	7327	16.92%
Medium	21645	50.00%
Low	14322	33.08%

The taxi demand by day in 2015 is shown below as a stacked bar chart. Since the timeseries signals are noisy, a power spectral density plot is included to detect if there are any noticeable frequencies underneath. From the result, no significant trend is found.



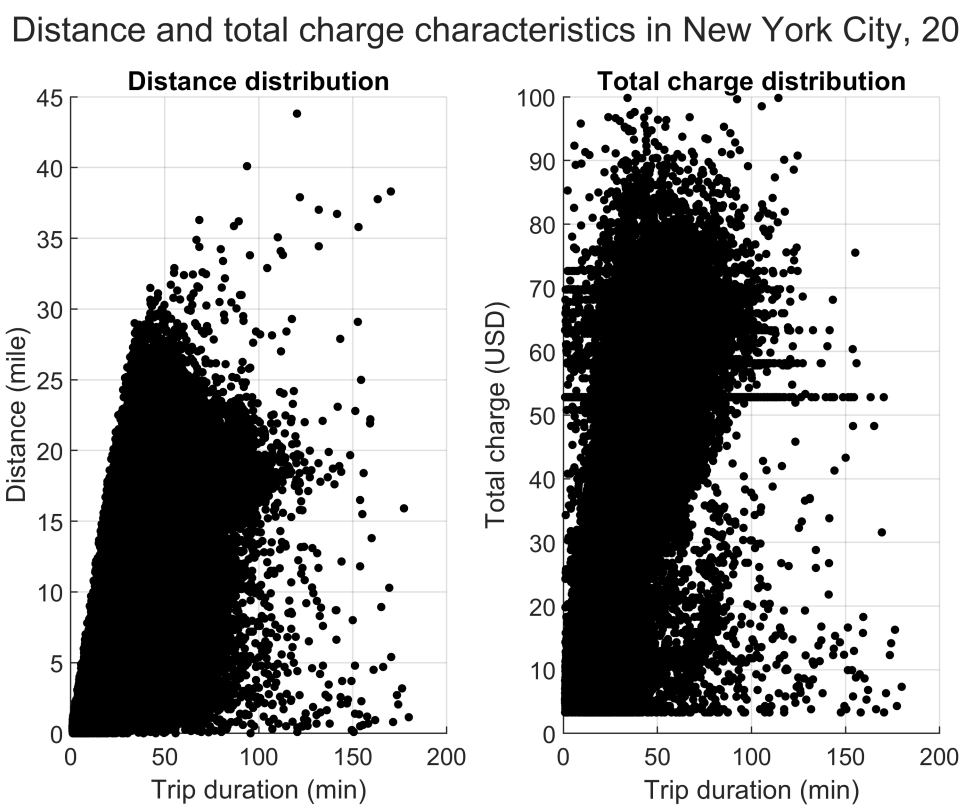
The Power Spectral Density (PSD) plot is included to detect frequencies underneath the noisy demand data. The results give no clue, either.



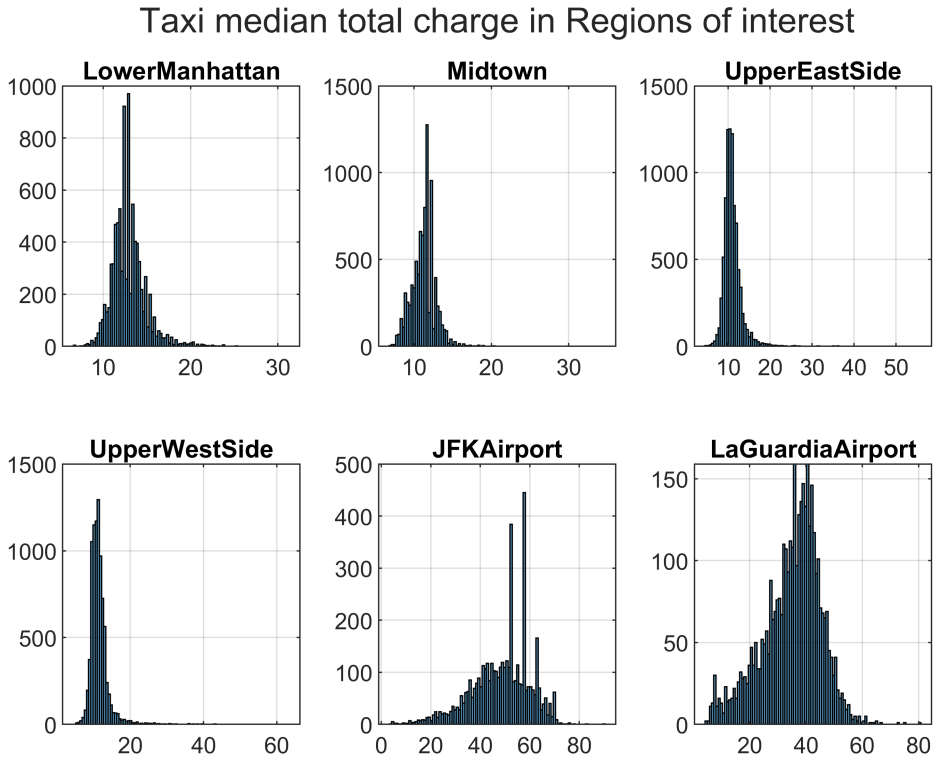
Overall exploration

To maximize profit, optimizing total charge is the top priority. Several attempts are made to find significant features related to total charge.

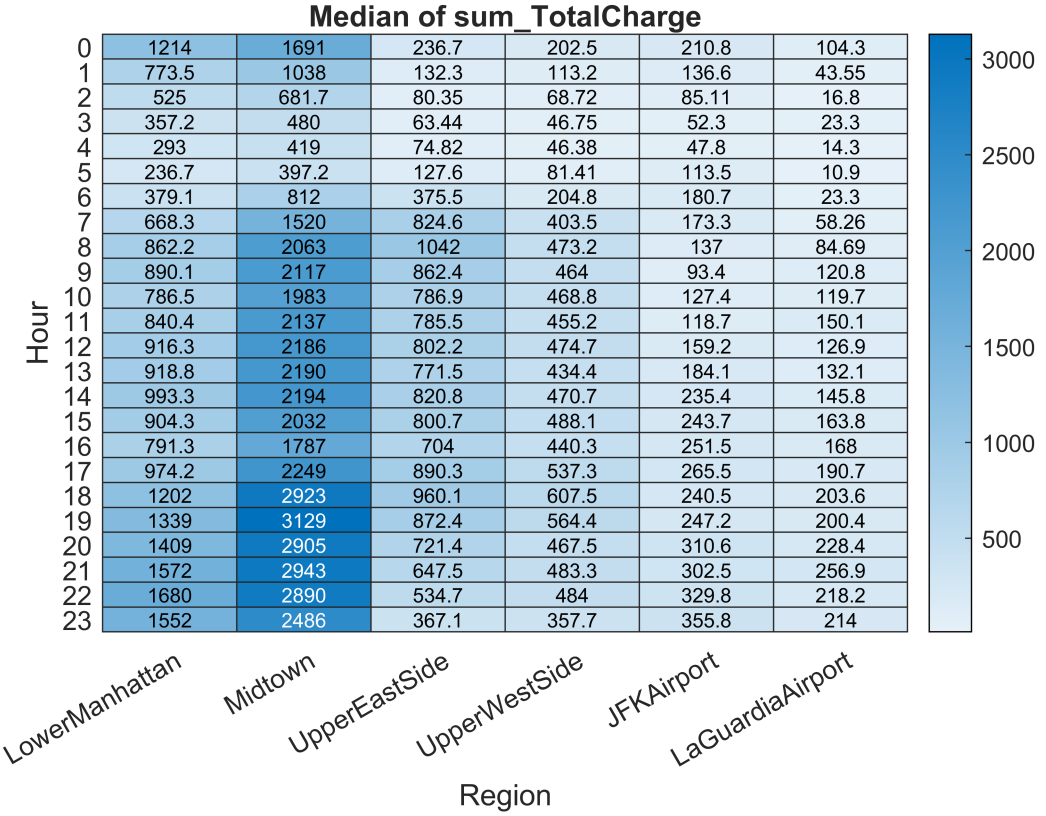
From the distance distribution, the distance is more likely to increase as the trip duration increases. From the total charge distribution, the total charge is more likely to increase as the trip duration increases. However, two clusters of data points exist at 52, and 60 USD. The author suspects the trend is affected by regions, so 6 histogram plots are provided for further insight.



From the histograms, the outliers come from JFK airport region. Upon investigation, they are the minimum charge for picking up at the airport.



The sum of total charge per demand is also an important factor for profit optimization. The heat map chart is plotted to show the effect from hour of the day and regions.



Hour	LowerManhattan	Midtown	UpperEastSide	UpperWestSide	JFKAirport	LaGuardiaAirport
0	1214	1691	236.7	202.5	210.8	104.3
1	773.5	1038	132.3	113.2	136.6	43.55
2	525	681.7	80.35	68.72	85.11	16.8
3	357.2	480	63.44	46.75	52.3	23.3
4	293	419	74.82	46.38	47.8	14.3
5	236.7	397.2	127.6	81.41	113.5	10.9
6	379.1	812	375.5	204.8	180.7	23.3
7	668.3	1520	824.6	403.5	173.3	58.26
8	862.2	2063	1042	473.2	137	84.69
9	890.1	2117	862.4	464	93.4	120.8
10	786.5	1983	786.9	468.8	127.4	119.7
11	840.4	2137	785.5	455.2	118.7	150.1
12	916.3	2186	802.2	474.7	159.2	126.9
13	918.8	2190	771.5	434.4	184.1	132.1
14	993.3	2194	820.8	470.7	235.4	145.8
15	904.3	2032	800.7	488.1	243.7	163.8
16	791.3	1787	704	440.3	251.5	168
17	974.2	2249	890.3	537.3	265.5	190.7
18	1202	2923	960.1	607.5	240.5	203.6
19	1339	3129	872.4	564.4	247.2	200.4
20	1409	2905	721.4	467.5	310.6	228.4
21	1572	2943	647.5	483.3	302.5	256.9
22	1680	2890	534.7	484	329.8	218.2
23	1552	2486	367.1	357.7	355.8	214

Relationship between Taxi Demand and Bank Holiday

From week 2 graded quiz, the author attempts to find the relationship between taxi demand and bank holiday. A flag `isHoliday` is included to check if a day of the year is bank holiday.

```
taxiSummaryHoliday = 43294×19 table
```

...

	Time_Date	Day	Hour	Region	PickupCount	DropoffCount
1	2015-01-01 00:...	1	0	LowerManhattan	56	41
2	2015-01-01 00:...	1	0	Midtown	118	72
3	2015-01-01 00:...	1	0	UpperEastSide	39	29
4	2015-01-01 00:...	1	0	UpperWestSide	23	16
5	2015-01-01 01:...	1	1	LowerManhattan	57	51
6	2015-01-01 01:...	1	1	Midtown	102	85
7	2015-01-01 01:...	1	1	UpperEastSide	40	40
8	2015-01-01 01:...	1	1	UpperWestSide	37	31
9	2015-01-01 02:...	1	2	LowerManhattan	57	51
10	2015-01-01 02:...	1	2	Midtown	101	95

	Time_Date	Day	Hour	Region	PickupCount	DropoffCount
11	2015-01-01 02:...	1	2	UpperEastSide	33	48
12	2015-01-01 02:...	1	2	UpperWestSide	23	16
13	2015-01-01 03:...	1	3	LowerManhattan	47	40
14	2015-01-01 03:...	1	3	Midtown	82	67
15	2015-01-01 03:...	1	3	UpperEastSide	21	34
16	2015-01-01 03:...	1	3	UpperWestSide	15	15
17	2015-01-01 04:...	1	4	LowerManhattan	44	26
18	2015-01-01 04:...	1	4	Midtown	66	41
19	2015-01-01 04:...	1	4	UpperEastSide	15	18
20	2015-01-01 04:...	1	4	UpperWestSide	6	4
21	2015-01-01 04:...	1	4	LaGuardiaAirp...	1	2
22	2015-01-01 05:...	1	5	LowerManhattan	18	14
23	2015-01-01 05:...	1	5	Midtown	39	31
24	2015-01-01 05:...	1	5	UpperEastSide	3	8
25	2015-01-01 05:...	1	5	UpperWestSide	6	6
26	2015-01-01 06:...	1	6	LowerManhattan	19	9
27	2015-01-01 06:...	1	6	Midtown	30	18
28	2015-01-01 06:...	1	6	UpperEastSide	6	12
29	2015-01-01 06:...	1	6	UpperWestSide	5	1
30	2015-01-01 06:...	1	6	JFKAirport	2	1
31	2015-01-01 07:...	1	7	LowerManhattan	14	7
32	2015-01-01 07:...	1	7	Midtown	27	28
33	2015-01-01 07:...	1	7	UpperEastSide	8	9
34	2015-01-01 07:...	1	7	UpperWestSide	3	3
35	2015-01-01 07:...	1	7	JFKAirport	2	2
36	2015-01-01 08:...	1	8	LowerManhattan	10	10
37	2015-01-01 08:...	1	8	Midtown	25	13
38	2015-01-01 08:...	1	8	UpperEastSide	7	4
39	2015-01-01 08:...	1	8	UpperWestSide	1	2
40	2015-01-01 08:...	1	8	LaGuardiaAirp...	1	2
41	2015-01-01 09:...	1	9	LowerManhattan	14	9
42	2015-01-01 09:...	1	9	Midtown	43	33
43	2015-01-01 09:...	1	9	UpperEastSide	9	8

	Time_Date	Day	Hour	Region	PickupCount	DropoffCount
44	2015-01-01 09:...	1	9	UpperWestSide	7	10
45	2015-01-01 09:...	1	9	JFKAirport	1	3
46	2015-01-01 09:...	1	9	LaGuardiaAirp...	3	2
47	2015-01-01 10:...	1	10	LowerManhattan	22	17
48	2015-01-01 10:...	1	10	Midtown	50	53
49	2015-01-01 10:...	1	10	UpperEastSide	21	8
50	2015-01-01 10:...	1	10	UpperWestSide	9	7
51	2015-01-01 10:...	1	10	LaGuardiaAirp...	4	1
52	2015-01-01 11:...	1	11	LowerManhattan	28	25
53	2015-01-01 11:...	1	11	Midtown	68	63
54	2015-01-01 11:...	1	11	UpperEastSide	27	23
55	2015-01-01 11:...	1	11	UpperWestSide	14	14
56	2015-01-01 11:...	1	11	JFKAirport	2	5
57	2015-01-01 11:...	1	11	LaGuardiaAirp...	3	2
58	2015-01-01 12:...	1	12	LowerManhattan	37	35
59	2015-01-01 12:...	1	12	Midtown	80	69
60	2015-01-01 12:...	1	12	UpperEastSide	28	24
61	2015-01-01 12:...	1	12	UpperWestSide	19	16
62	2015-01-01 12:...	1	12	JFKAirport	4	1
63	2015-01-01 12:...	1	12	LaGuardiaAirp...	3	2
64	2015-01-01 13:...	1	13	LowerManhattan	45	40
65	2015-01-01 13:...	1	13	Midtown	97	98
66	2015-01-01 13:...	1	13	UpperEastSide	22	19
67	2015-01-01 13:...	1	13	UpperWestSide	16	16
68	2015-01-01 13:...	1	13	JFKAirport	4	3
69	2015-01-01 13:...	1	13	LaGuardiaAirp...	8	6
70	2015-01-01 14:...	1	14	LowerManhattan	35	37
71	2015-01-01 14:...	1	14	Midtown	77	82
72	2015-01-01 14:...	1	14	UpperEastSide	31	27
73	2015-01-01 14:...	1	14	UpperWestSide	18	14
74	2015-01-01 14:...	1	14	LaGuardiaAirp...	3	6
75	2015-01-01 15:...	1	15	LowerManhattan	52	50
76	2015-01-01 15:...	1	15	Midtown	87	75

	Time_Date	Day	Hour	Region	PickupCount	DropoffCount
77	2015-01-01 15:...	1	15	UpperEastSide	33	32
78	2015-01-01 15:...	1	15	UpperWestSide	21	22
79	2015-01-01 15:...	1	15	JFKAirport	3	2
80	2015-01-01 15:...	1	15	LaGuardiaAirp...	3	2
81	2015-01-01 16:...	1	16	LowerManhattan	32	29
82	2015-01-01 16:...	1	16	Midtown	72	86
83	2015-01-01 16:...	1	16	UpperEastSide	30	25
84	2015-01-01 16:...	1	16	UpperWestSide	13	16
85	2015-01-01 16:...	1	16	JFKAirport	5	1
86	2015-01-01 17:...	1	17	LowerManhattan	38	32
87	2015-01-01 17:...	1	17	Midtown	92	92
88	2015-01-01 17:...	1	17	UpperEastSide	32	34
89	2015-01-01 17:...	1	17	UpperWestSide	30	22
90	2015-01-01 17:...	1	17	JFKAirport	5	4
91	2015-01-01 17:...	1	17	LaGuardiaAirp...	7	3
92	2015-01-01 18:...	1	18	LowerManhattan	38	35
93	2015-01-01 18:...	1	18	Midtown	84	92
94	2015-01-01 18:...	1	18	UpperEastSide	27	30
95	2015-01-01 18:...	1	18	UpperWestSide	14	13
96	2015-01-01 18:...	1	18	JFKAirport	3	1
97	2015-01-01 19:...	1	19	LowerManhattan	30	35
98	2015-01-01 19:...	1	19	Midtown	79	71
99	2015-01-01 19:...	1	19	UpperEastSide	25	28
100	2015-01-01 19:...	1	19	UpperWestSide	13	11

⋮

The cross tabulation determines if day of the year and demand are independent. The p-value for the test `pChi = 0.9079` suggests, at 95% confidence level, we cannot reject the null hypothesis that day of the year and demand are independent. In other words, day of the year and demand are highly correlated.

```
chi2 = 677.8571
pChi = 0.9079
```

The ANOVA test determines if high, medium, and low demand by day are correlated. The p-value for the test `pAnova = 0.002` suggests, at 95% confidence level, we can reject the null hypothesis that high, medium, and low demand by day are independent of each other.

```
pAnova = 0.0020
```

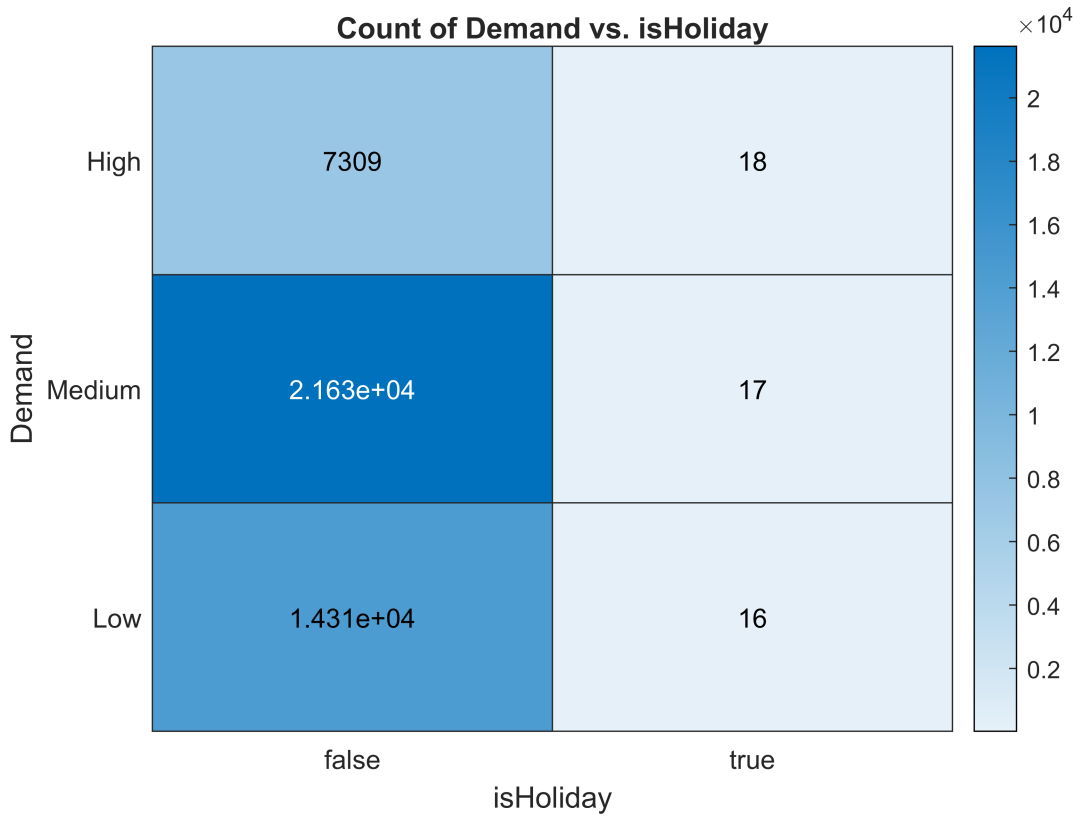
tbl = 4×6 cell

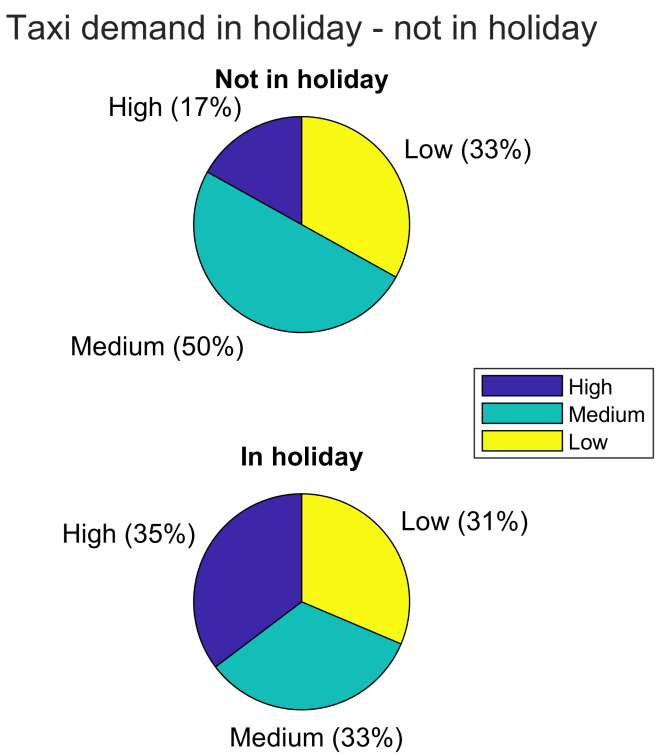
	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'F'	'Prob>F'
2	'Groups'	139120	2	69558	6.2257	0.002
3	'Error'	483680000	43291	11173	[]	[]
4	'Total'	483820000	43293	[]	[]	[]

In summary, the p-value for the test `pHoliday = 0.0015` suggests, at 95% confidence level, we can reject the null hypothesis that Taxi demand and Bank holidays are independent. In other words, the taxi demand and bank holidays are highly correlated.

```
chi2 = 13.0620
pHoliday = 0.0015
```

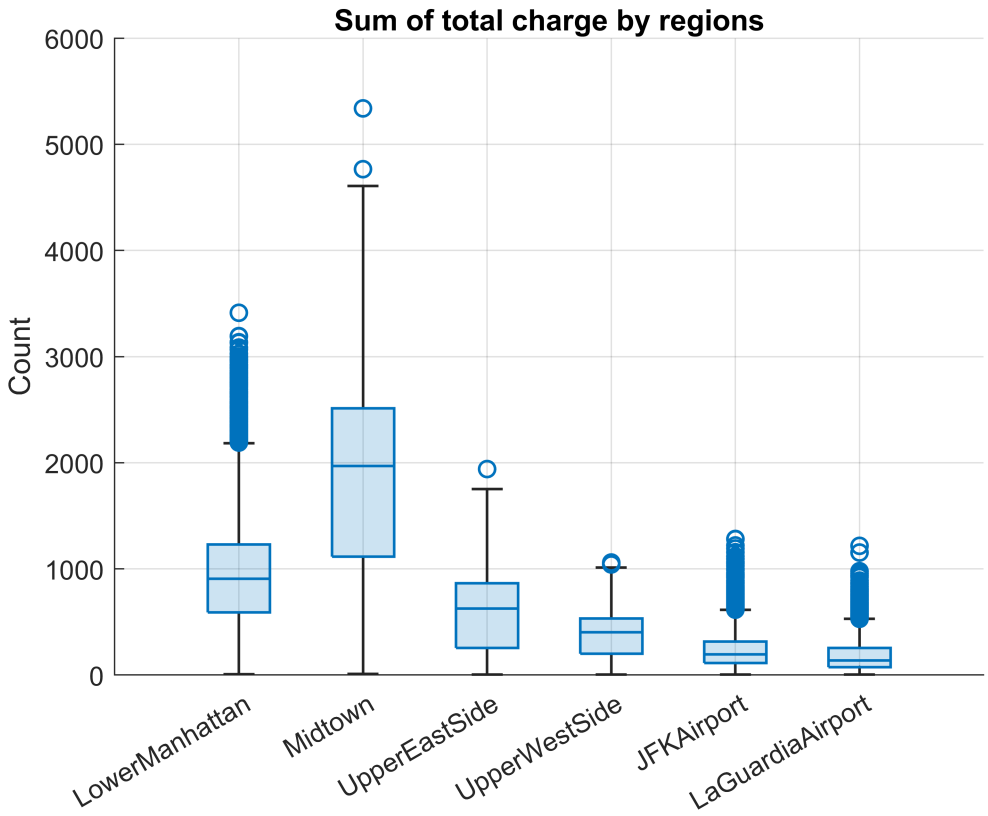
The results are illustrated in the heat map and pie charts below:





Relationship between Demand and Total charge

It is easy to assume that demand rises will lead to increased revenue by intuition. However, showing the correlation between two variables are desireable because it is proven by data. The box chart below shows the interquartile range of 6 regions of interest.

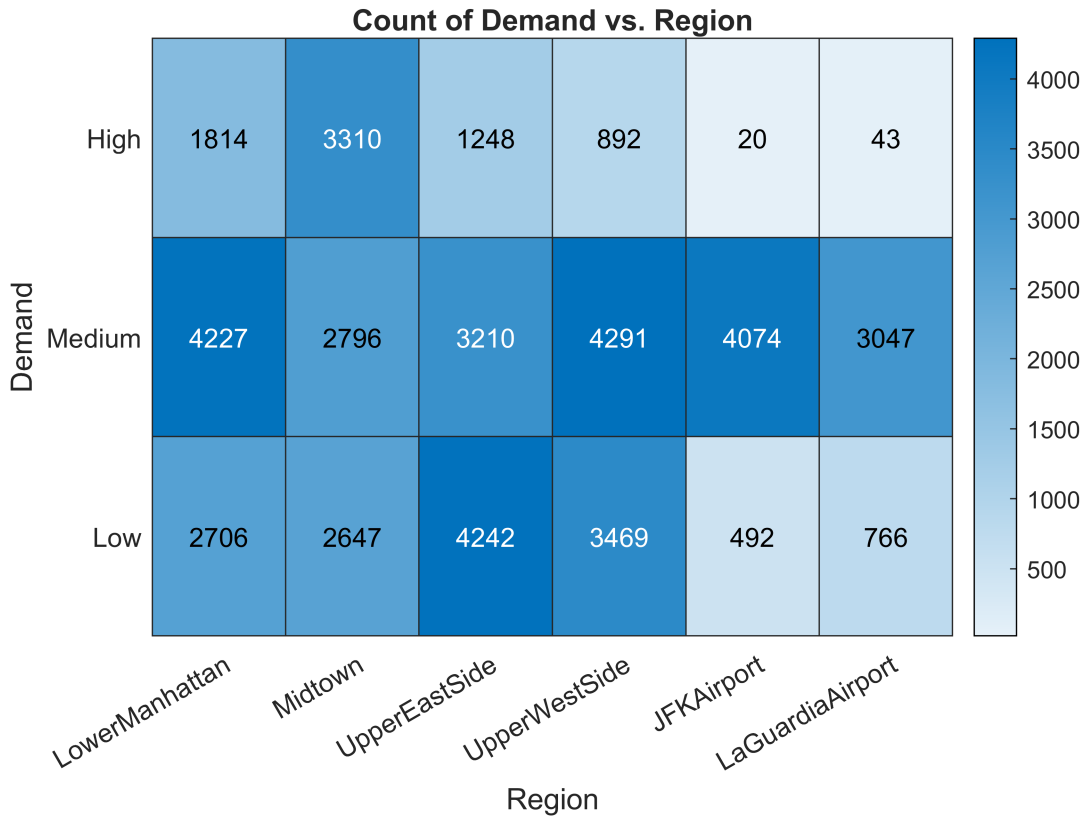


The Kruskal-Wallis test determines if high, medium, and low demand have the same distribution. The p-value for the test `p = 0.0` suggests, at 95% confidence level, we can reject the null hypothesis that high, medium, and low demand come from the same distribution. In other words, the distributions of 3 demand types are different.

```
p = 0
tbl = 4×6 cell
```

	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'Chi-sq'	'Prob>Chi-sq'
2	'Groups'	14396000...	2	71979000...	9216.3	0
3	'Error'	53228000...	43291	122950000	[]	[]
4	'Total'	67624000...	43293	[]	[]	[]

The figure below shows how frequent each demand type occurs in 6 regions of interest in 2015. The medium demand is the highest, followed by low demand. In 2 airport regions, the medium demand is dominant.



Apply the Supervised Machine Learning Workflow

From a taxi driver's perspective, the predictors for demand are hour of a day, day of the year, bank holidays, and the region he intends to pickup/dropoff customers. However, Super Taxis can provide more predictors since the organization stores invaluable amount of data to extract useful features for training accurate predictive machine learning models. By using past data to predict future demands, additional predictors are the means and medians of distance, duration, and total charge in 2015. In total, the predictive model will use 10 features, four of which are recognized by the drivers, and the remaining six are included from Super Taxis' data in 2015.

Create test data

In machine learning implementation, the data from table `taxiSummaryHoliday` is divided into two groups. 80% of the data is stored in `taxiTrain` variable, which is used for training the predictive model. The remaining 20% is stored in `taxiTest` variable, which is used for model validation.

Train the models

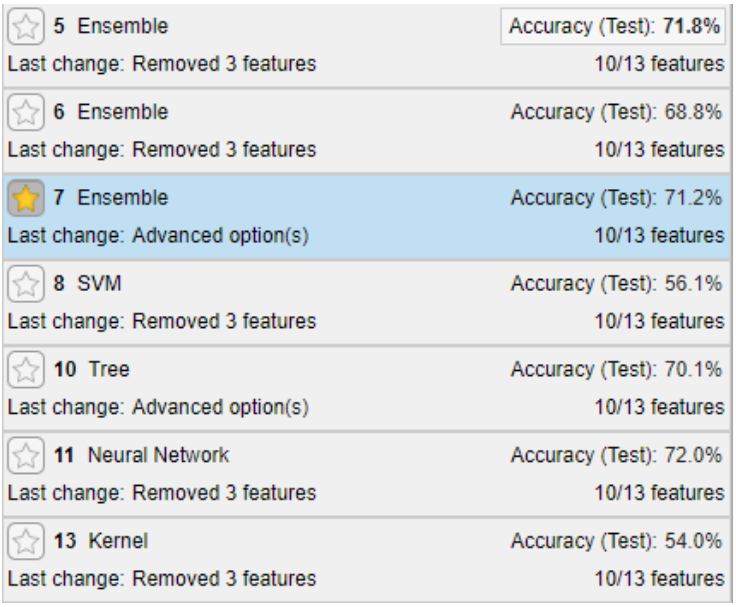
7 models were examined

- **Model 5**: optimized Ensemble Classifier model, type bagged trees, default cost matrix. In general, the algorithm provides accurate prediction under minor pertubations. However, it is unstable against outliers in the training dataset.
- **Model 6**: optimized Ensemble Classifier model, type RUSBoosted trees, custom cost matrix. The RUSBoosting option manipulates the training dataset.

- **Model 7**: optimized Ensemble Classifier model, type bagged trees, custom cost matrix. The bagging option manipulates the training dataset.
- **Model 8**: optimized Support Vector Machine (SVM) model, default cost matrix. In general, the algorithm provides very accurate prediction and robust to noise. However, it is computationally expensive to train and difficult to handle missing values.
- **Model 10**: optimized Decision Trees model, default cost matrix. In general, the algorithm provides moderately accurate prediction and it is computationally less expensive to train than SVM model.
- **Model 11**: Neural Network Classifier model, type Trilayered Neural Network, cost matrix unavailable. Since the Neural Network Classifier model cannot evaluate miscalculation costs, it can only be used as the baseline model for comparison.
- **Model 12**: Kernel Approximation Classifier model, type Logistic Regression Kernel, default cost matrix. The algorithm provides accurate prediction if the response variables gather in a cluster. However, it is very computationally expensive.

The criteria for model selection are:

- Good accuracy
- Able to predict such that the output correctly matches the regions with highest demand, all the while minimizing taxi deployment to regions with low demand.



Out of 7 models, 2 models were selected for representing Scenario 1 and 2:

1. **Scenario 1**: Model 5 is implemented as the baseline model that emphasizes overall accuracy. This model will be useful for analysis and comparison later. The model types and hyperparameters, class imbalance, and significant features are redundant for simplicity.
2. **Scenario 2**: Model 7 is chosen with modified miscalculation cost. After training, the model will be compared with Model 5 to assess the reduction in prediction accuracy.

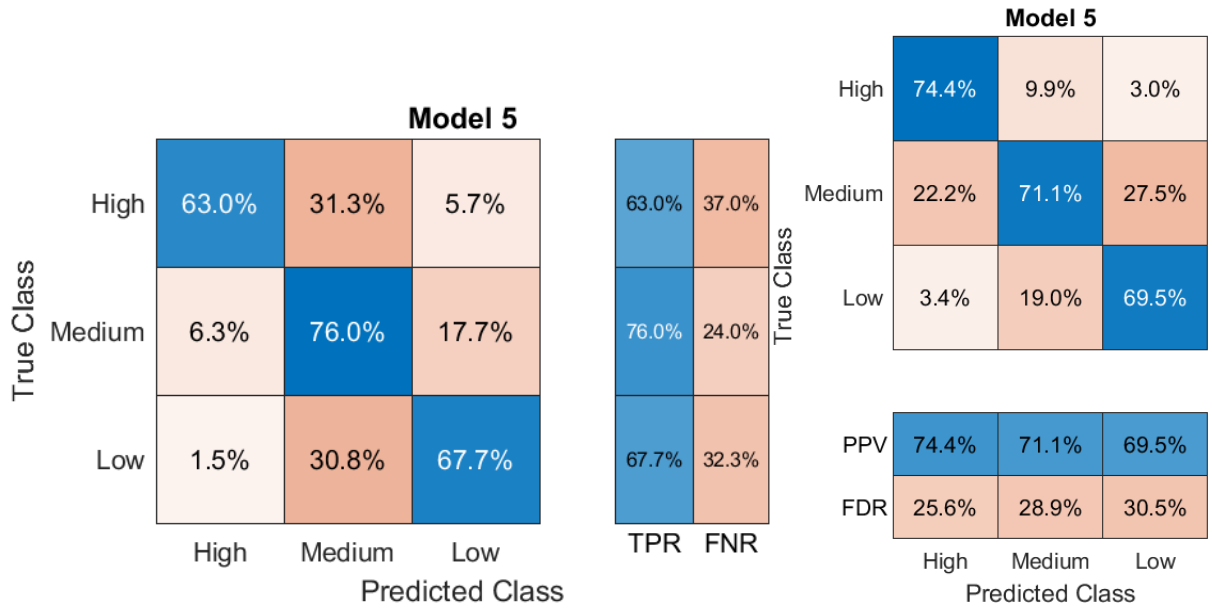
Scenario 1: Default miscalculation cost

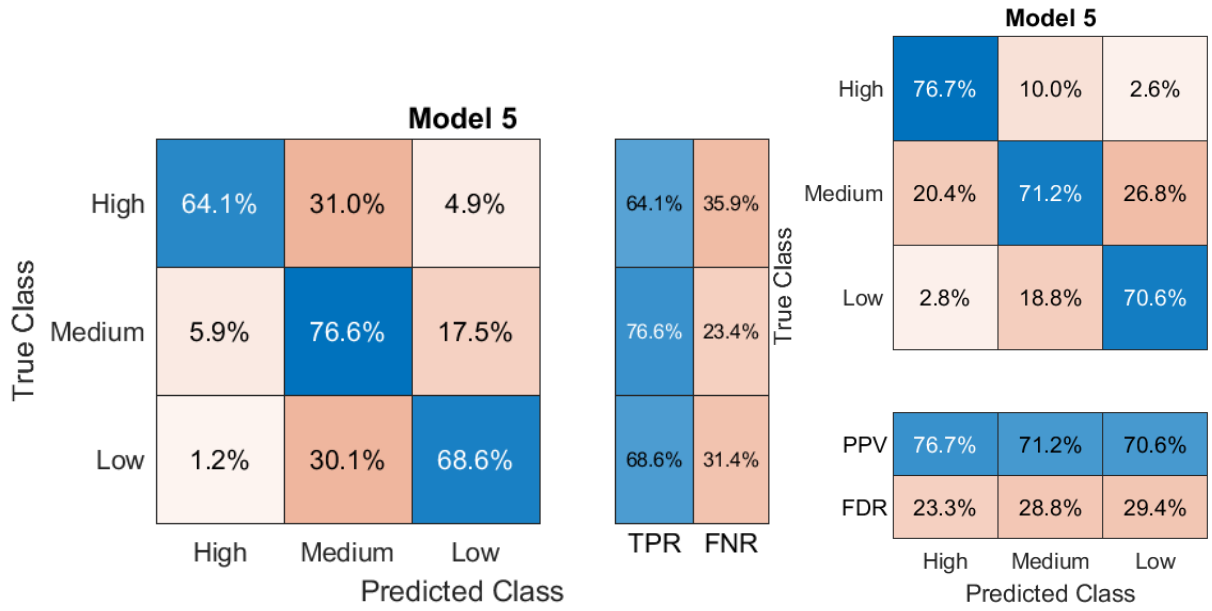
The trained model was validated using cross-validation method at 5 folds. The hyperparameters of the bagged decision tree model were determined using Bayesian optimization. The acquisition function of the Bayesian Optimization Algorithm uses type "Expected improvement per second plus" for 30 iterations without training time limit. Principle Component Analysis (PCA) was disabled, and 10 features are used. The features were:

- Hour of the day, and day of the year (2 features).
- mean value of duration, distance, and total charge (3 features).
- median value of duration, distance, and total charge (3 features).
- Pickup/dropoff region, and a flag variable for bank holiday (2 features)

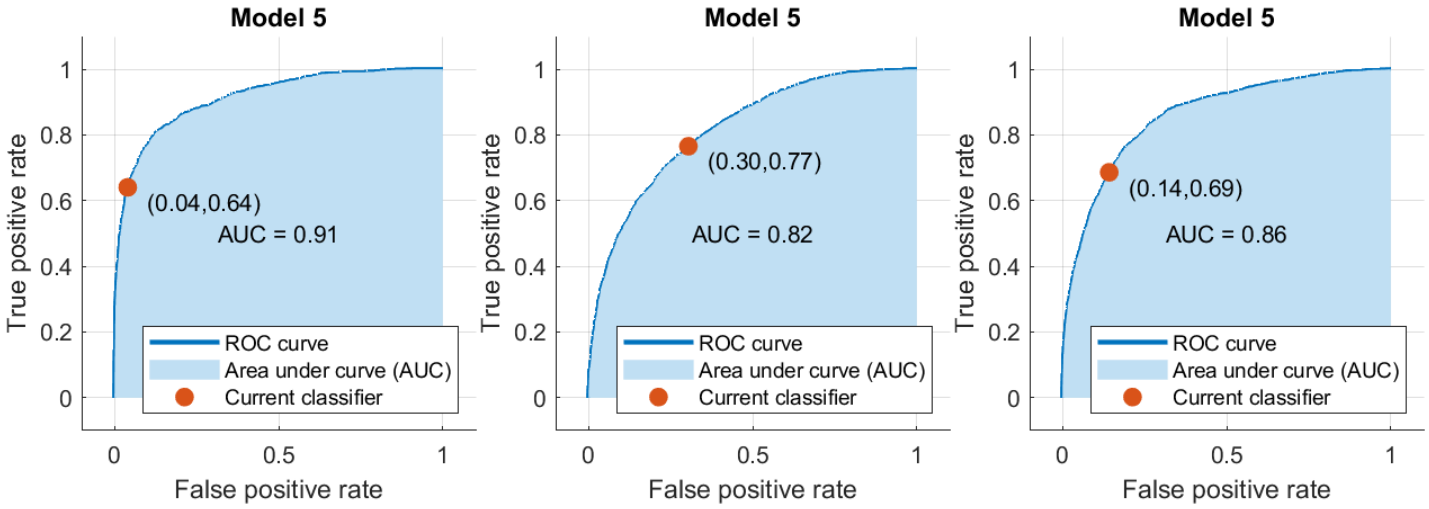
After training for 20 minutes, the model is tested in Classification Learner App with test data `taxiTest` as input. The validation results show the prediction speed at around 21,000 observations per second. Compared to the validation results, the accuracy of the output is 71.8%, increased from 71.1%. The total cost of the test result is 2,442, decreased from 10,020.

The confusion matrix shows the characteristics of the model. True Positive Rate (TPR), False Negative Rate (FNR), Positive Predictive Value (PPV), False Detection Rate (FDR) are the diagnostic results. The matrices above are the results of the model using 5 fold cross-validation, while the matrices below are the results of the model using `taxiTest` as the test data.





The figures below show the Receiver Operating Characteristic (ROC) curve of the model after testing using `taxiTest` as the test data. An ROC curve is a graphical plot used to show the diagnostic ability of binary classifiers. It is constructed by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The more AUC value closer to 1, the more accurate the prediction.



Scenario 2: Customized miscalculation cost

The criteria for taxi deployment strategy are:

- Always go to the nearest High demand region when one is available.
- Go to the nearest Medium demand region if there is no High demand region available.
- Never go to or stay in a Low demand region.

		Predicted Class		
		High	Medium	Low
True Class	High	0	4	6
	Medium	7	0	3
	Low	10	5	0

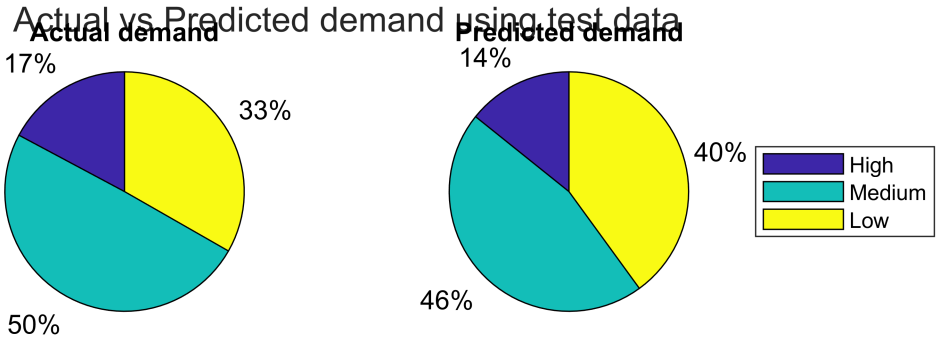
The figure shows the modified cost matrix for Model 7. In Scenario 2, the misclassification costs are quantified in accordance with the requirements. The main points are repeated below:

- Emphasize reducing false negatives for **Low** demand, especially **Low** demand classified as **High** demand.
- Also, try to reduce false positives for **Low**, especially missed **High**.
- Some increase in false positives for **Low** demand is an acceptable trade-off to reduce false negatives for **Low** demand.
- **Medium**/**High** demand misclassification is not a priority to reduce.
- Use reasonable modifications to the cost matrix to achieve your goals.
- Use your test data set to verify your modeling results.

```
myModel =
  ClassificationBaggedEnsemble
             PredictorNames: {'Day'  'Hour'  'Region'  'mean_Distance'  'median_Distance'  'mean_Duration'  'median_Du
               ResponseName: 'Y'
      CategoricalPredictors: [3 10]
                 ClassNames: [High    Medium    Low]
             ScoreTransform: 'none'
            NumObservations: 34636
                 NumTrained: 480
                     Method: 'Bag'
               LearnerNames: {'Tree'}
     ReasonForTermination: 'Terminated normally after completing the requested number of training cycles.'
                  FitInfo: []
       FitInfoDescription: 'None'
                 FResample: 1
                   Replace: 1
          UseObsForLearner: [34636×480 logical]


   Properties, Methods
```

After training the predictive model using ensemble bagged trees classification approach, the test data is used to validate the result. At first glance, the model predicts moderately close to the actual result, illustrated by the pie charts below:

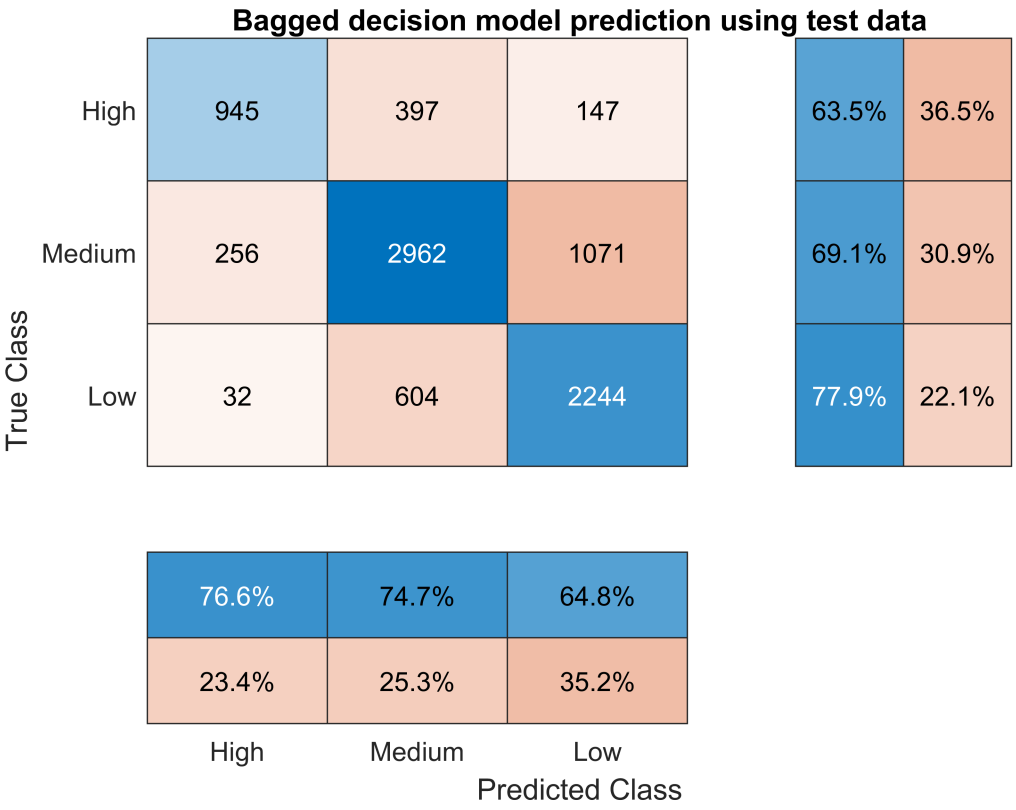


The classification metrics evaluate how well the predictive model perform.

```
Accuracy = 71.04%
cResult = 5×5 table
```

	Precision	Recall	Fallout	Specificity	F1
1 High	0.7664	0.6347	0.0402	0.9598	0.6943
2 Medium	0.7474	0.6906	0.2291	0.7709	0.7179
3 Low	0.6482	0.7792	0.2108	0.7892	0.7077
4 Avg	0.7207	0.7015	0.1600	0.8400	0.7066
5 WgtAvg	0.7177	0.7104	0.1905	0.8095	0.7104

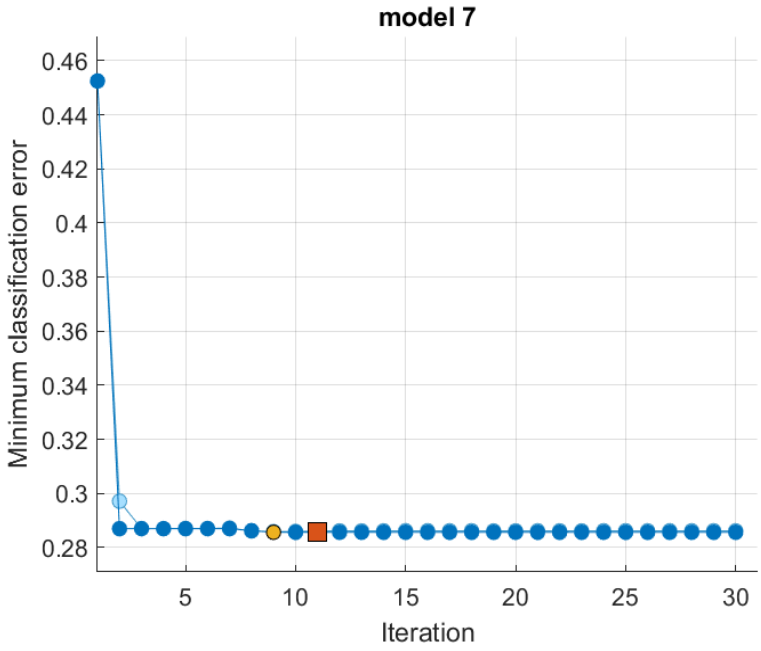
Model 7 produces the following confusion matrix. Out of 8,658 demand data points, only 32 cases fall into the critical situation, which accounts for 0.37%. These cases were listed as unwanted because deploying the taxi fleet to low demand regions would result in traffic jam and high failure cost.



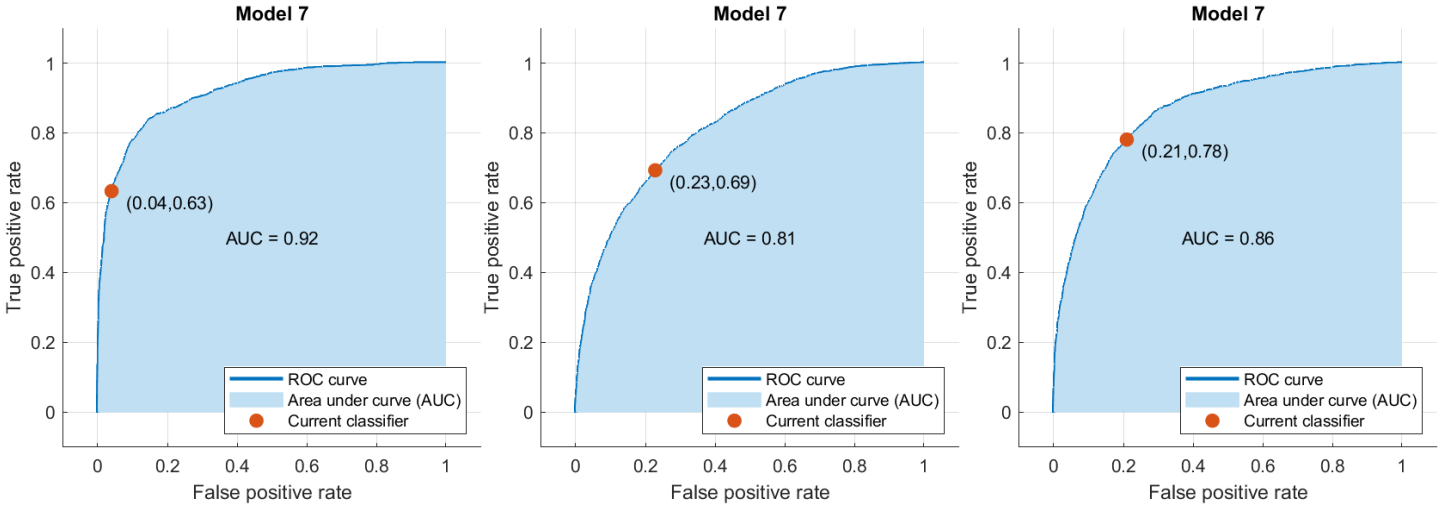
The trained model was validated using cross-validation method at 5 folds. The hyperparameters of the bagged decision tree model were determined using Bayesian optimization. The acquisition function of the Bayesian Optimization Algorithm uses type "Expected improvement per second plus" for 30 iterations without training time limit. Principle Component Analysis (PCA) was disabled, and 10 features are used. The features were:

- Hour of the day, and day of the year (2 features).
- mean value of duration, distance, and total charge (3 features).
- median value of duration, distance, and total charge (3 features).
- Pickup/dropoff region, and a flag variable for bank holiday (2 features)

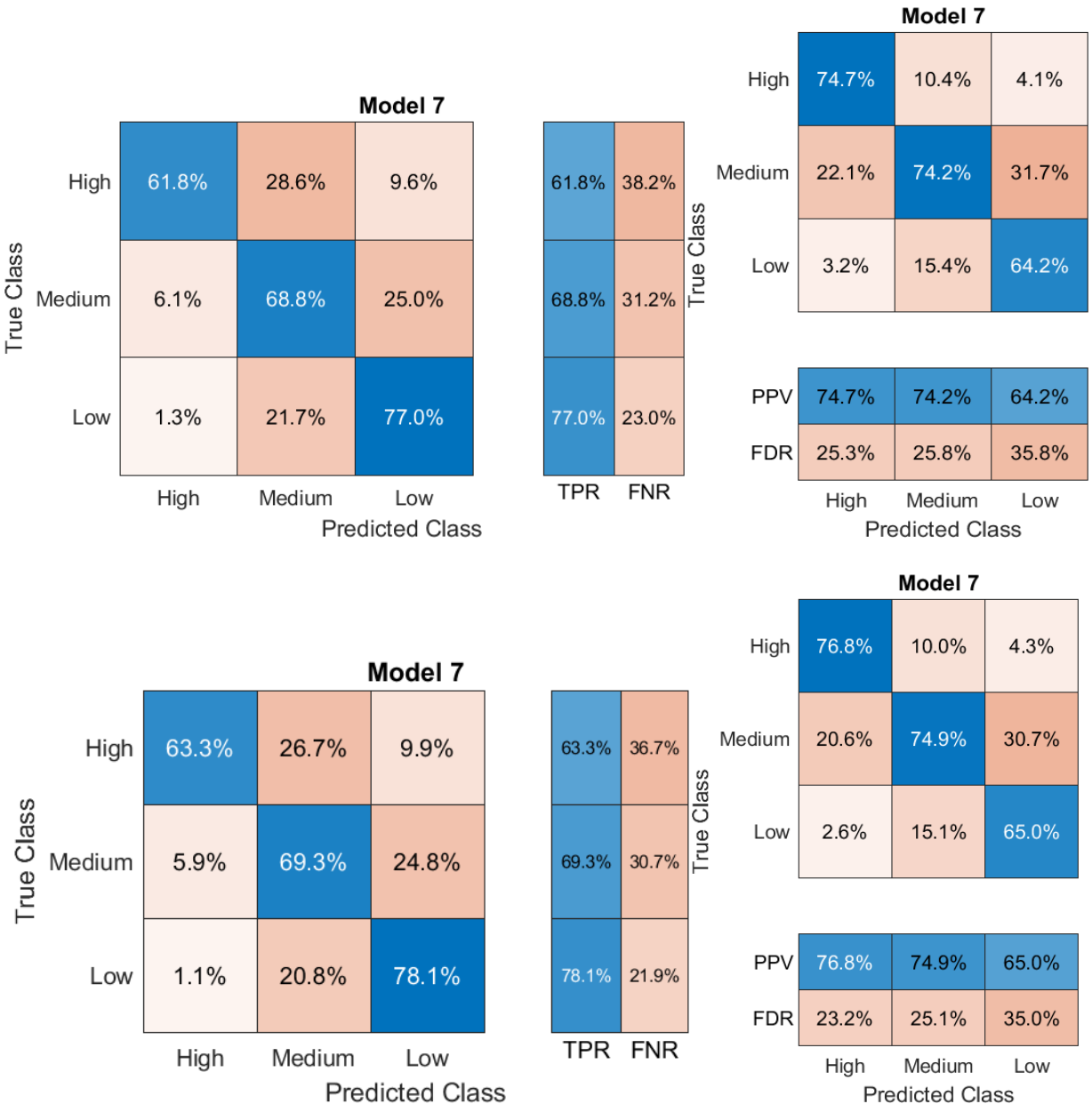
After training for 30 minutes and 30 iterations (shown below), the model is tested in Classification Learner App with test data `taxiTest` as input. The validation results show the model predicts slower than that of Scenario 1, at around 5,400 observations per second. Compared to the validation results, the accuracy of the output is 71.2%, increased from 70.3%. The total cost of the test result is 10,758, decreased from 44,473.



The figures below show the Receiver Operating Characteristic (ROC) curve of the model after testing using `taxiTest` as the test data. An ROC curve is a graphical plot used to show the diagnostic ability of binary classifiers. It is constructed by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The more AUC value closer to 1, the more accurate the prediction.From left to right, the figures show ROC curves of high, medium, and low demand, respectively.



The confusion matrix shows the characteristics of the model. True Positive Rate (TPR), False Negative Rate (FNR), Positive Predictive Value (PPV), False Detection Rate (FDR) are the diagnostic results. The matrices above are the results of the model using 5 fold cross-validation, while the matrices below are the results of the model using `taxiTest` as the test data.



Conclusions

After hypothesizing and validating with statistical tests, the significant features which helps predict demand are:

- Hour of the day, and day of the year (2 features).
- mean value of duration, distance, and total charge (3 features).
- median value of duration, distance, and total charge (3 features).
- Pickup/dropoff region, and a flag variable for bank holiday (2 features)

After implement machine learning to predict the demand accuracy, the results are as follows:

- The standard predictive model yields 71.8% accuracy.
- When the cost for false prediction is taken into account, the model yields 71.2% accuracy.

Appendix

```
function [trainedClassifier, validationAccuracy] = trainClassifierModifiedCM(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% Returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: A table containing the same predictor and response
%       columns as those imported into the app.
%
%  Output:
%      trainedClassifier: A struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: A function to make predictions on new
%       data.
%
%      validationAccuracy: A double containing the accuracy as a
%       percentage. In the app, the Models pane displays this overall
%       accuracy score for each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 17-Feb-2023 20:30:26


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Day', 'Hour', 'Region', 'mean_Distance', 'sum_Distance', 'median_Distance', 'mean_Duration'
predictors = inputTable(:, predictorNames);
response = inputTable.Demand;
isCategoricalPredictor = [false, false, true, false, false, false, false, false, false, false, false, false, t
```

```
% Data transformation: Select subset of the features
% This code selects the same subset of features as were used in the app.
includedPredictorNames = predictors.Properties.VariableNames([true true true true false true true false true t
predictors = predictors(:,includedPredictorNames);
isCategoricalPredictor = isCategoricalPredictor([true true true true false true true false true true false tru

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateTree(...
    'MaxNumSplits', 722, ...
    'NumVariablesToSample', 'all');
classificationEnsemble = fitcensemble(...
    predictors, ...
    response, ...
    'Method', 'Bag', ...
    'NumLearningCycles', 480, ...
    'Learners', template, ...
    'Cost', [0 4 6; 7 0 3; 10 5 0], ...
    'ClassNames', categorical({'High'; 'Medium'; 'Low'}, {'High' 'Medium' 'Low'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
featureSelectionFcn = @(x) x(:,includedPredictorNames);
ensemblePredictFcn = @(x) predict(classificationEnsemble, x);
trainedClassifier.predictFcn = @(x) ensemblePredictFcn(featureSelectionFcn(predictorExtractionFcn(x)));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'Day', 'Hour', 'Region', 'isHoliday', 'mean_Distance', 'mean_Duration',
trainedClassifier.ClassificationEnsemble = classificationEnsemble;
trainedClassifier.About = 'This struct is a trained model exported from Classification Learner R2021b.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new table, T, use: \n  yfit = c.predictFcn(

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Day', 'Hour', 'Region', 'mean_Distance', 'sum_Distance', 'median_Distance', 'mean_Duration'
predictors = inputTable(:, predictorNames);
response = inputTable.Demand;
isCategoricalPredictor = [false, false, true, false, false, false, false, false, false, false, false, false, t

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationEnsemble, 'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

Quiz results

Week 1

Practice quiz

```
q1 = 2922266
```

```
q2 = 10
```

```
q3 = 7×3 table
```

	PickupRegion	GroupCount	median_Distance
1	JFKAirport	62178	17.5300
2	LaGuardiaAirport	70720	9.7300
3	Others	176146	2.0500
4	LowerManhattan	580591	2
5	UpperWestSide	265523	1.6900
6	Midtown	1296074	1.5400
7	UpperEastSide	422235	1.5000

```
q4 = 0.9230
```

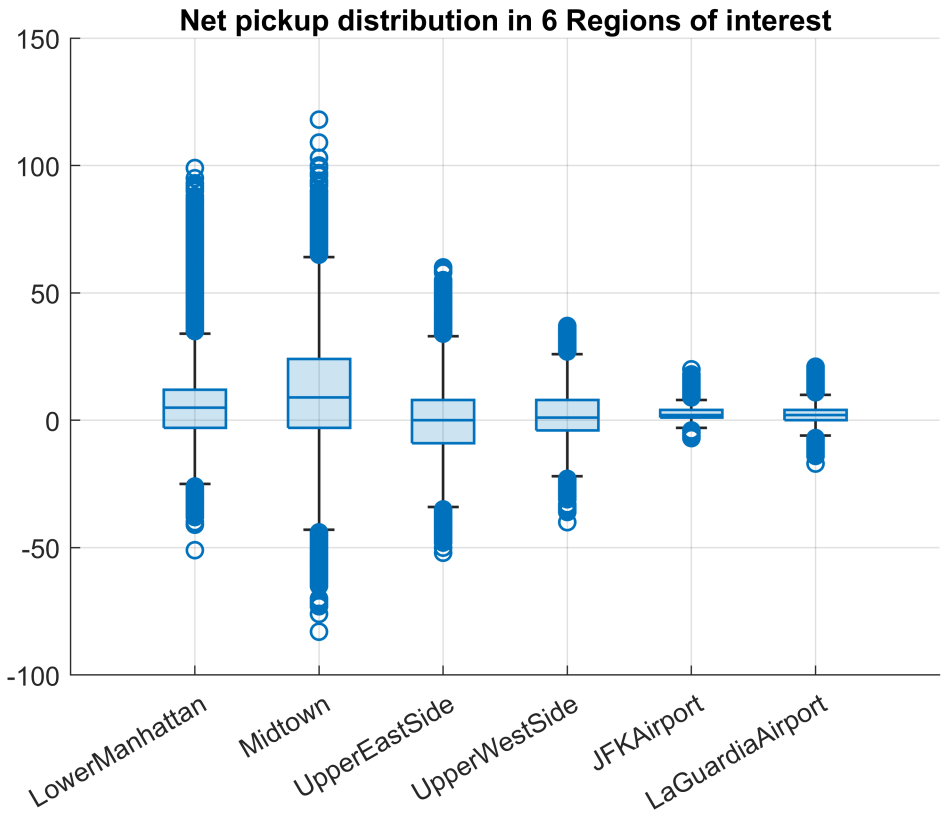
Graded quiz

```
q2 = 0.0722
```

```
q3 = 121
```

```
q4 = 8×3 table
```

	PickupRegion	GroupCount	mean_Fare
1	JFKAirport	35064	41.8832
2	LaGuardiaAirport	32912	29.0103
3	<undefined>	1065	13.5053
4	Others	164875	12.3966
5	LowerManhattan	565386	12.1548
6	Midtown	1243158	10.6753
7	UpperWestSide	257904	10.2787
8	UpperEastSide	411015	10.1913



q6 = 6×3 table

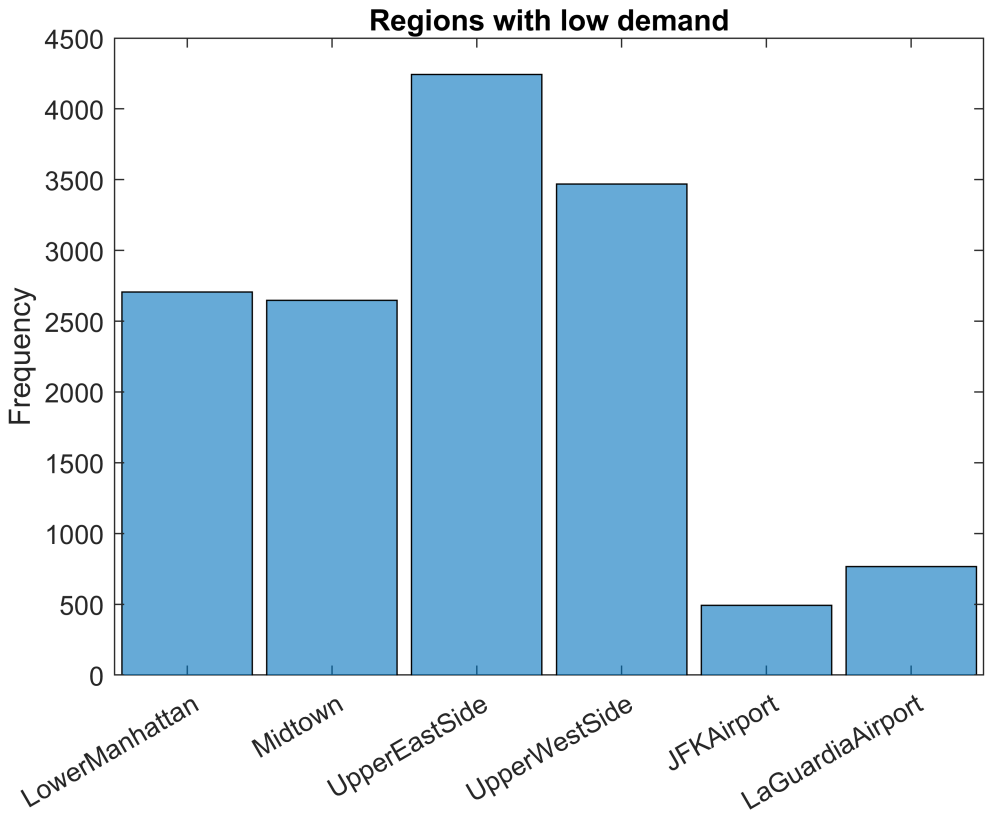
	Region	GroupCount	Correlation
1	Midtown	8753	-0.1387
2	LowerManhattan	8747	-0.0154
3	UpperEastSide	8700	0.1859
4	UpperWestSide	8652	0.4376
5	JFKAirport	4586	0.5637
6	LaGuardiaAirport	3856	0.6096

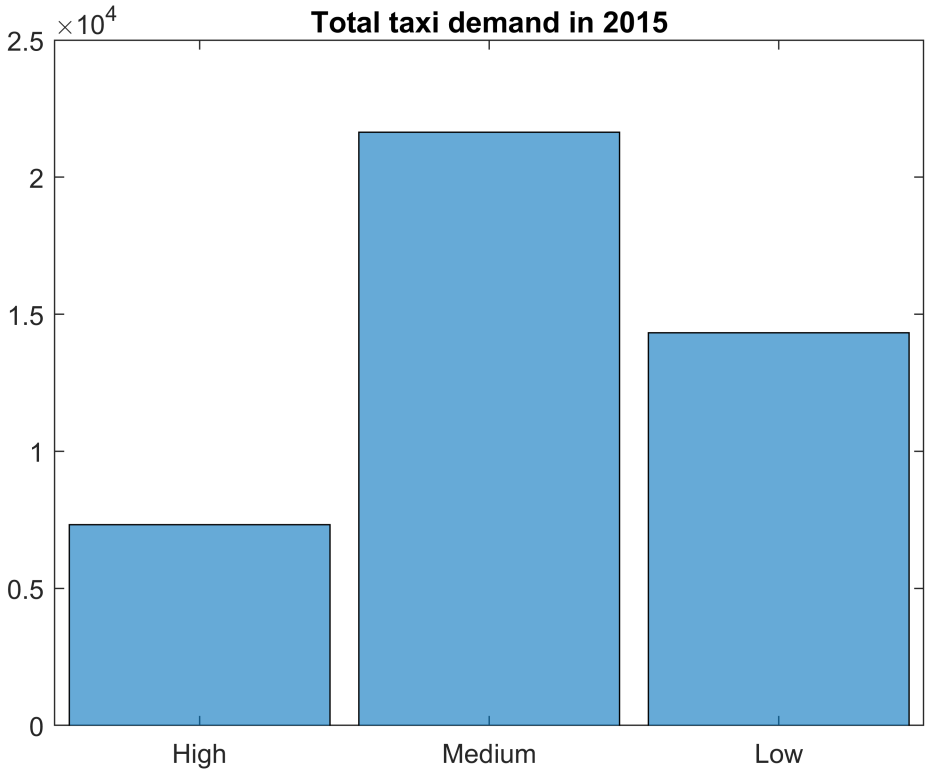
Week 2

Practice quiz

```
q1 = 17

     High          7327
     Medium       21645
     Low          14322
```





Graded quiz

```
q2 = 0.9079
```

```
q3 = 0.0020
```

```
q5 = 0.0015
```

Week 3

Practice quiz

```
q1 = 17
```

```
q2 = 0.9079
```

```
q3 = 0.0020
```

Graded quiz

```
q1 = 17
```

```
q2 = 0.9079
```

```
q3 = 0.0020
```