

---

---

# 모빌리티(헤일링) 서비스를 위한 택시수요예측모델

DSFT01 오세광

---



# 1. 목차

- 모빌리티 서비스의 특징
- 데이터 전처리 & 시각화
- 택시 수요예측 모델 제작
- 프로젝트 인사이트

# 모빌리티 서비스 특징?



1. 왜 모빌리티 서비스에 주목하는가?
2. 프로젝트에서 예측하고자 하는 것은?

## 카카오 택시 이용 이유 및 계기 (중복응답)

[출처:트렌드모니터 / 단위:%]






## 모빌리티 서비스 특징

- 택시서비스는 택시 이용 전 서비스의 품질을 알기 어려운 속성을 지닌 '경험재'에 해당하며, 이용자 입장에서 서비스 품질에 대한 불확실성이 높다.
- 택시 서비스는 얇은 시장으로 택시의 수요는 이용자 위치에 따라 지리적으로 분산되어 있어 공급과 수요의 매칭에 어려움이 있다.

출처: \* 참조 :2020 카카오 모빌리티 리포트  
<https://brunch.co.kr/@kakaomobility/58>, 18~28p

출처 : <https://transportkuu.com/2020/05/20/%ED%83%9D%EC%8B%9C-%EB%AC%B8%EC%A0%9C/>







## 국내 모빌리티 삼국지

회사명	최근 투자 유치	사용자 규모
 카카오 모빌리티	칼라일그룹으로부터 2억달러 투자 유치	‘카카오T’ 가입자 2800만명
 T맵 모빌리티	우버로부터 5000만달러 투자. 이후 합작사에 1억 달러 이상 투자 약속	‘T맵’ 월간 활성이용자 1300만명
 소카	SG 프라이빗에쿼티 등으로부터 600억원 투자 유치	회원수 640만명

자료= 각 사

출처 : [https://www.chosun.com/economy/tech\\_it/2021/02/27/OL7RYHNXRRDRFICFUZXWRYC2QY/](https://www.chosun.com/economy/tech_it/2021/02/27/OL7RYHNXRRDRFICFUZXWRYC2QY/)

## 주요 기업 가맹택시 추진 현황

						
기업	카카오 모빌리티	KST모빌리티	쏘카(VCNC)	코나투스	우버	포티투닷
서비스	카카오T 블루	마카롱택시	타다 라이트	반반택시 그린	미정	유모스텝
운영 대수	1만372대 (9월 기준)	1만600대 (9월 기준)	500대 이상 (모집 중)	1,000대 (10월 말 예정)	사업 검토 중, SKT와 손잡음	사업 검토 중 (미정)
특징	1위 호출 앱과 카카오 프렌즈 캐릭터 활용, 콜비 최대 3,000원	콜비 1,000원, 1만대 업계 최초 돌파	10월 말 출시, 타다 앱과 브랜드 이미지 활용	9월 말 전주, 10월 말 서울 예정, 코로나19 보험 적용·태블릿 설치 등	미정	미정

출처 : <https://www.mk.co.kr/news/it/view/2020/10/1058745/>

# 모빌리티 서비스 특징

- 모빌리티 데이터를 활용한 카카오T 블루, 타다라이트, 우버, 그랩과 같은 헤일링 서비스(ride hailing service)가 기존의 모빌리티 서비스(택시)를 개선할 수 있는 가능성을 보여주고 있다.
- 기존의 불편한 택시 서비스를 개선하여 이용자에게 헤일링 서비스를 통해 고품질의 서비스를 제공할 수 있다.
- 제도적인 접근이 아닌 기술적인 접근으로 기존 서비스에 문제점을 해결할 실마리를 찾을 수 있다.

카카오 T 택시 운행 과정별 기술적 진전과 활용 데이터

운행단계	과정	기술 기반 혁신	활용 데이터
운행전 단계	호출	이용자 니즈 분석에 기반한 다양한 서비스	예상 요금, 호출 가능 차량 데이터 등
	배차	인공지능 기반 배차 알고리즘	기사평가, 기사 배차 수락율, 기사 운행 패턴, 택시 수공급비, 실시간 교통상황, 최근 운행 분포, ETA (예상 소요 시간) 등
운행중 단계	픽업	GPS 기반 이용자 위치 및 목적지 확인, 기사 프로필 확인	이용자 / 기사 위치 데이터, 기사 / 차량 데이터 (성함, 사진, 차량정보, 위치, 연락처 등), 소요시간, 이동거리, 도로정보 등
	주행	최적 경로 선정, 앱미터기 기반 요금 산정	교통정보, 실시간 위치 데이터, 요금정보 (기본 요금, 시간 요금, 거리 요금, 시간거리 병산 요금, 할증요금) 등
운행후 단계	결제	자동결제, 다양한 요금제	이용자 인증 데이터, 결제 매체 데이터 등
	평가	택시 품질 평가, 이용자 평가	이동경로, 친절 태고 데이터 등

## 예측하고자 하는 것은?

- 특정시간대에 특정 장소에서의 택시 수요를 예측하고자 한다.
- 택시 수요의 불균등한 분포를 수요 예측지점에 따라 택시 공급조정할 수 있다.
- 수요와 공급을 매칭을 시킴으로써 얇은 시장에서 두꺼운 시장으로 전환할 수 있다.

출처: \* 참조 :2020 카카오 모빌리티 리포트  
<https://brunch.co.kr/@kakaomobility/58>, 18~28p



# 데이터 전처리 & 시각화



1. 뉴욕 택시 데이터는 어떻게 구성되어 있는가?
2. 필요 없는 데이터는 어떻게 제거할 것인가?
3. 그 데이터들을 어떻게 시각화 해볼 것인가?

# 뉴욕 택시 데이터

	VendorID	tpet_pickup_datetime	tpet_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RateCodeID	s
0	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	1.59	-73.993896	40.750111	1	
1	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	3.30	-74.001648	40.724243	1	
2	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	1.80	-73.963341	40.802788	1	
3	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	0.50	-74.009087	40.713818	1	
4	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	3.00	-73.971176	40.762428	1	
...	...	...	...	...	...	...	...	...	...
12748981	1	2015-01-10 19:01:44	2015-01-10 19:05:40	2	1.00	-73.951988	40.786217	1	
12748982	1	2015-01-10 19:01:44	2015-01-10 19:07:26	2	0.80	-73.982742	40.728184	1	
12748983	1	2015-01-10 19:01:44	2015-01-10 19:15:01	1	3.40	-73.979324	40.749550	1	
12748984	1	2015-01-10 19:01:44	2015-01-10 19:17:03	1	1.30	-73.999565	40.738483	1	
12748985	1	2015-01-10 19:01:45	2015-01-10 19:07:33	1	0.70	-73.960350	40.766399	1	

12748986 rows × 19 columns

This data dictionary describes yellow taxi trip data. For a dictionary describing green taxi data, or a map of the TLC Taxi Zones, please visit [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml).

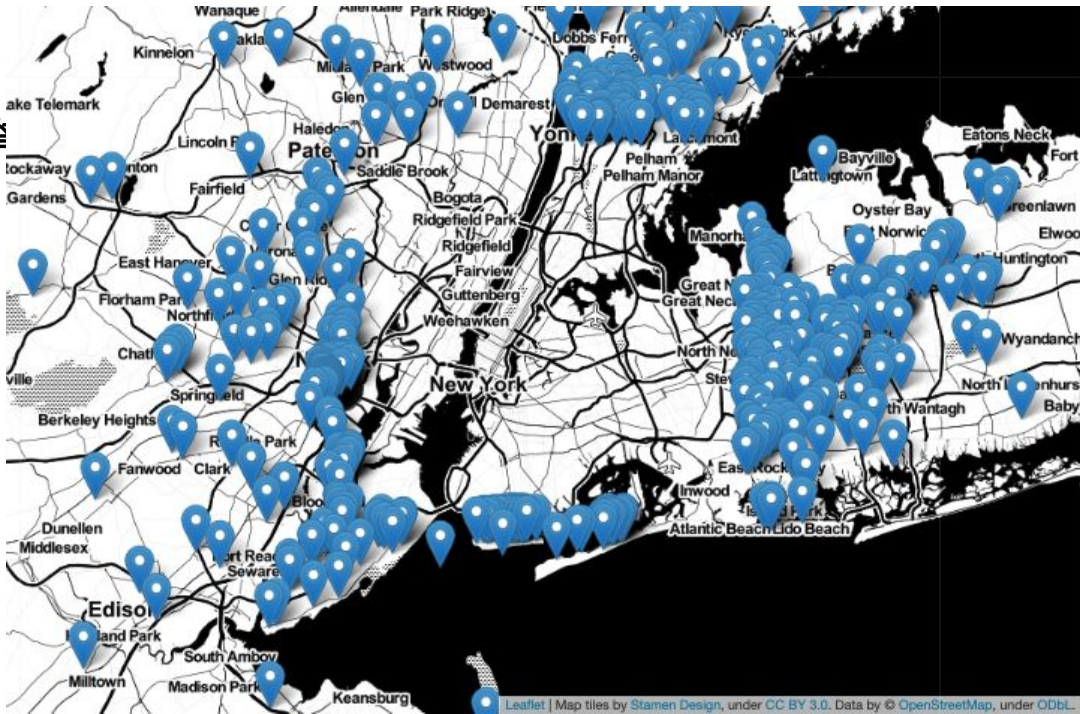
Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record.  1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle.  This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip.  1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server.  Y= store and forward trip N= not a store and forward trip
Payment_type	A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the \$0.50 and \$1 rush hour and overnight charges.
MTA_tax	\$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	\$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.

# 뉴욕 택시데이터

- 픽업 시간/ 도착 시간
- 운행거리
- 운영요금
- 승객 수
- 픽업 / 도착 위치정보
- 등

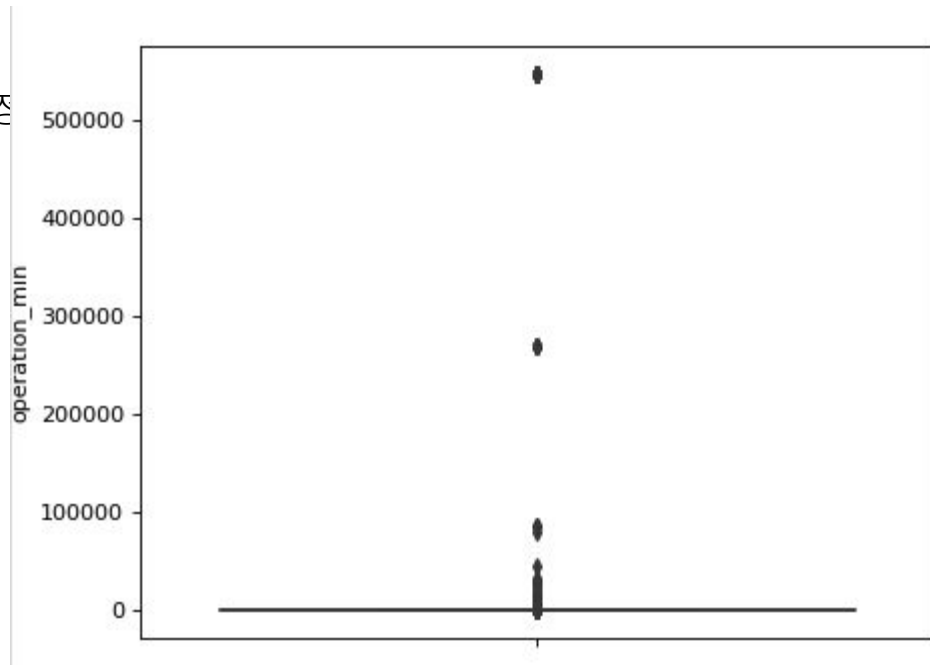
# 데이터 전처리

- 뉴욕시 위도와 경도 확인 & 범위 설정
- 탑승시각 이상치 제거
- 택시 속도 이상치 제거
- 운행거리 이상치 제거
- 택시요금 이상치 제거



# 데이터 전처리

- 뉴욕시 위도와 경도 확인 & 범위 설정
- 탑승시각 이상치 제거
- 택시 속도 이상치 제거
- 운행거리 이상치 제거
- 택시요금 이상치 제거



# 데이터 전처리

- 뉴욕시 위도와 경도 훑기
- 탑승시각 이상치 제거
- 택시 속도 이상치 제거
- 운행거리 이상치 제거
- 택시요금 이상치 제거

```
1 ## 스피드 체크 디테일하게
2 for i in np.arange(0.0,1.0,0.1): # 90%에서 디테일하게 보기
3     var = df_taxi_operation_time_modified['speed'].values
4     var = np.sort(var, axis=None)
5
6     print("{} %는 {}".format(99+i, var[int(len(var)*(float(99+i)/100))]))
7 print("100%는", var[-1])
```

99.0 %는 35.75286041189931  
99.1 %는 36.312364425162684  
99.2 %는 36.91672401927048  
99.3 %는 37.59036144578313  
99.4 %는 38.332225913621265  
99.5 %는 39.17826825127335  
99.6 %는 40.15655577299413  
99.7 %는 41.342756183745585  
99.8 %는 42.87162162162162  
99.9 %는 45.31858407079647  
100%는 192857142.85714284

# 데이터 전처리

- 뉴욕시 위도와 경도 후
- 탑승시각 이상치 제거
- 택시 속도 이상치 제거
- 운행거리 이상치 제거
- 택시요금 이상치 제거

```
1 for i in np.arange(0.0,1.0,0.1):
2     var = df_taxi_operation_time_modified['trip_distance'].values
3     var = np.sort(var, axis=None)
4     print("{} %는 {}".format(99+i, var[int(len(var)*(float(99+i)/100))]))
5 print("100%는",var[-1])
```

```
99.0 %는 18.17
99.1 %는 18.37
99.2 %는 18.6
99.3 %는 18.83
99.4 %는 19.13
99.5 %는 19.5
99.6 %는 19.96
99.7 %는 20.5
99.8 %는 21.22
99.9 %는 22.57
100%는 258.9
```

# 데이터 전처리

- 뉴욕시 위도와 경도 후
- 탑승시각 이상치 제거
- 택시 속도 이상치 제거
- 운행거리 이상치 제거
- 택시요금 이상치 제거

```
1 ## 한번 보기
2 for i in np.arange(0.0,1.0,0.1):
3     var = df_taxi_operation_time_modified['total_amount'].values
4     var = np.sort(var, axis=None)
5     print("{} %는 {}".format(99+i, var[int(len(var)*(float(99+i)/100))]))
6 print("100%는",var[-1])
```

```
99.0 %는 66.13
99.1 %는 68.13
99.2 %는 69.6
99.3 %는 69.6
99.4 %는 69.73
99.5 %는 69.75
99.6 %는 69.76
99.7 %는 72.58
99.8 %는 75.35
99.9 %는 88.28
100%는 3950611.6
```



# 데이터 전처리

- 함수로 아웃라이어 정리

#지금까지 데이터 시각화를 통해 확인한 아웃라이어들을 정리하는 함수를 만든다.

```
def remove_outliers(df_taxi_modified):  
    data_of_number = df_taxi_modified.shape[0]  
    print("선택한 데이터의 수", data_of_number)
```

```
1 print("2015년 1월 제거된 아웃라이어")  
2 print("-----")  
3 df_taxi_data_outliers_removed = remove_outliers(df_taxi_modified)  
4 print("아웃라이어 제거 후에 남겨진 데이터의 비율은", float(len(df_taxi_data_outliers_removed))/len(df_taxi_modified))
```

2015년 1월 제거된 아웃라이어

-----

선택한 데이터의 수 12748986

뉴욕지역을 벗어나는 아웃라이어의 수는 293919

택시운행시간을 분석해봤을 때 아웃라이어의 수는 23889

택시운행거리를 분석해봤을 때 아웃라이어의 수는 92597

택시운행속도 분석해봤을 때 아웃라이어의 수는 101830

택시요금 분석해봤을 때 아웃라이어의 수는 5275

총 제거된 아웃라이어의 수는 377910

-----

아웃라이어 제거 후에 남겨진 데이터의 비율은 0.9703576425607495

# 택시 수요예측 모델 제작



1. 어떻게 지역들을  
분할할 것인가?

2. 분할지역을 시간별로  
어떻게 나눌 것인가?

3. 수요예측모델  
성능은?

# 서브 지역으로 분할

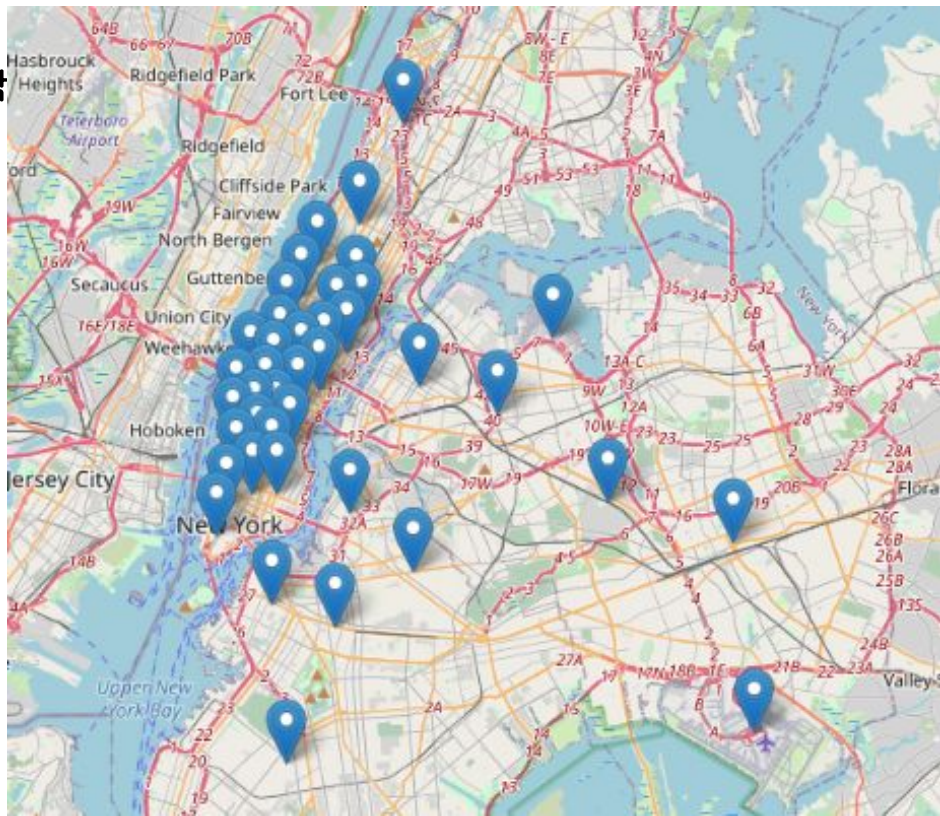
- 서브 지역에서 수요
- 클러스터링을 통한 분할
- K값을 변화로 최적화 찾기

```
1 for increment in range(10, 100, 10):  
2     cluster_centers, cluster_len = find_clusters(increment)  
3     find_min_distance(cluster_centers, cluster_len)
```

```
On choosing a cluster size of 10  
Avg. Number of Clusters within the vicinity (i.e. intercluster-distance < 2): 2.0  
Avg. Number of Clusters outside the vicinity (i.e. intercluster-distance > 2): 8.0  
Min inter-cluster distance = 1.0945442325142662  
---  
On choosing a cluster size of 20  
Avg. Number of Clusters within the vicinity (i.e. intercluster-distance < 2): 4.0  
Avg. Number of Clusters outside the vicinity (i.e. intercluster-distance > 2): 16.0  
Min inter-cluster distance = 0.7131298007388065  
---  
On choosing a cluster size of 30  
Avg. Number of Clusters within the vicinity (i.e. intercluster-distance < 2): 8.0  
Avg. Number of Clusters outside the vicinity (i.e. intercluster-distance > 2): 22.0  
Min inter-cluster distance = 0.5185088176172186  
---  
On choosing a cluster size of 40  
Avg. Number of Clusters within the vicinity (i.e. intercluster-distance < 2): 8.0  
Avg. Number of Clusters outside the vicinity (i.e. intercluster-distance > 2): 32.0  
Min inter-cluster distance = 0.5069768450365043  
---  
On choosing a cluster size of 50  
Avg. Number of Clusters within the vicinity (i.e. intercluster-distance < 2): 12.0  
Avg. Number of Clusters outside the vicinity (i.e. intercluster-distance > 2): 38.0  
Min inter-cluster distance = 0.36536302598358383  
---
```

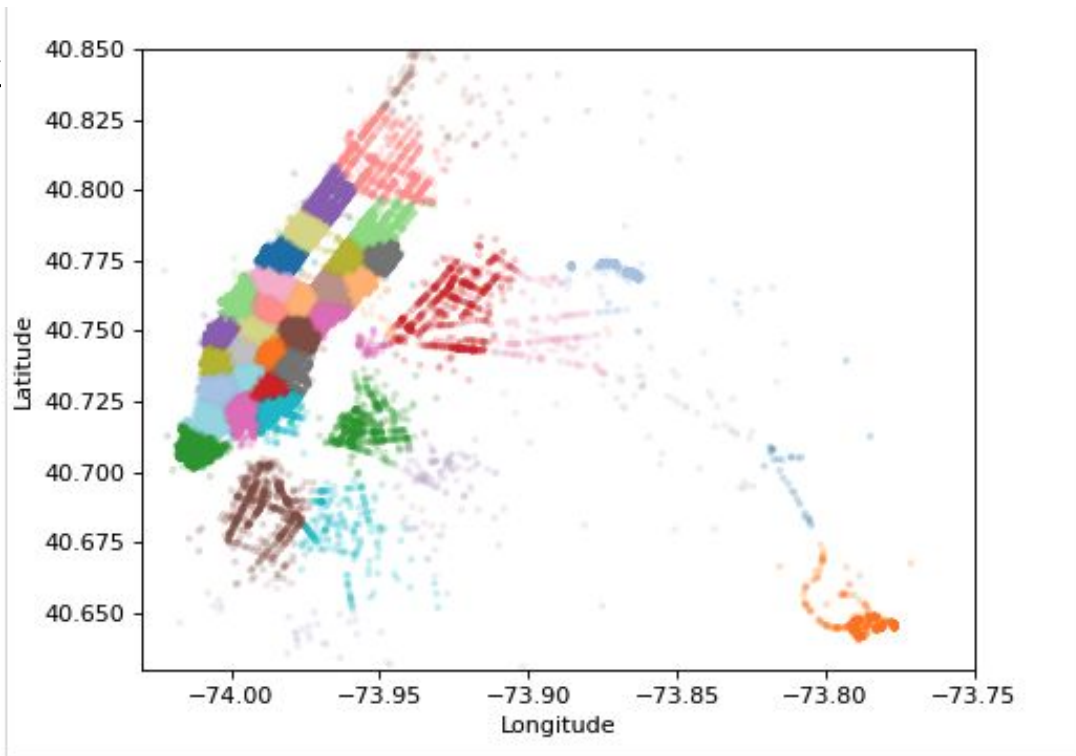
# 서브 지역으로 분할

- 서브 지역에서 수요가 얼마인지 파악
- 클러스터링을 통한 분할
- 40개 지역의 중심지역



# 서브 지역으로 분할

- 서브 지역에서 수요가 얼마인지
- 클러스터링을 통한 분할
- 40개 지역의 분포



# 시간대 별로 데이터 구분

- 10분단위로 데이터들을 구분
- $(24\text{시간} \times 60\text{분} / 10\text{분}) \times 40\text{time bin}$
- 40개 지역의 분포지역

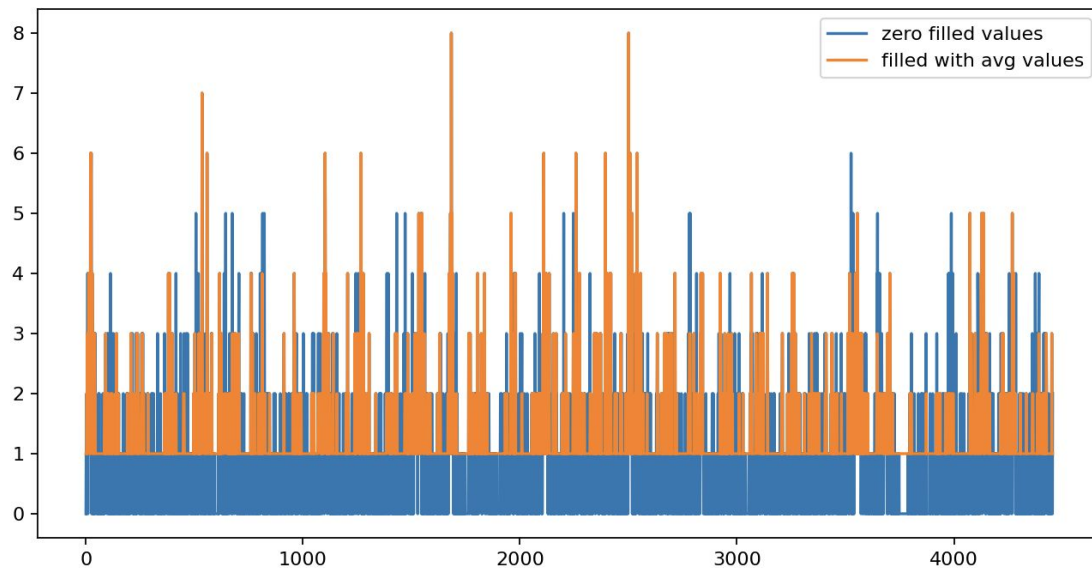
```
1 jan_2015_groupby.head()
```

		trip_distance
pickup_cluster	pickup_bins	
0	0	104
	1	200
	2	208
	3	141
	4	155

total_amount	operation_min	speed	pickup_times	pickup_cluster	pickup_bins
17.05	18.050000	5.285319	1.421316e+09	34	2130
17.80	19.833333	9.983193	1.420890e+09	2	1419
10.80	10.050000	10.746269	1.420890e+09	16	1419
4.80	1.866667	16.071429	1.420890e+09	38	1419
16.30	19.316667	9.318378	1.420890e+09	22	1419

# 평활화

- missing data에 0으로 채움
- missing data에 평균값으로 채움
- 예측 효율을 높여주기 위
- 큰 트렌드를 맞출 수 있음



# 예측 기본 모델

- pickup ratio :  $R_t = 2016\text{년 수요량} / 2015\text{년 수요량}$
- 작년 시간 대에 비해 얼마가 증감할 것인지 예측
- $R_{t+1} * \text{작년 탑승자 수}$
- simple moving average / weighted moving average  
/exponential weighted moving average



# 예측 기본 모델

- **simple moving average** : 단순 평균
- **weighted moving average** : 가까운 시간 대에는 **weight**를 더 주는 방식
- **exponential weighted moving average** : 가까운 시간 대에는 기하승수적으로 **weight**를 더 주는 방식

# 예측 기본 모델

- **MAPE** : 몇 퍼센트의 오차가 있는지?
- **MSE** : 평균 오차가 어느 정도인지?
- 기본 모델이기 때문에 그렇게 성능이 좋지 않았다.

Error Metric Matrix (Forecasting Methods) - MAPE & MSE

Moving Averages (Ratios) -	MAPE: 0.22050823069428171	MSE: 1020.1420642921147
Moving Averages (2016 Values) -	MAPE: 0.15655221491950697	MSE: 279.0884464605735
Weighted Moving Averages (Ratios) -	MAPE: 0.21899644827798298	MSE: 981.7608254928315
Weighted Moving Averages (2016 Values) -	MAPE: 0.1488018659974338	MSE: 244.06921482974911
Exponential Moving Averages (Ratios) -	MAPE: 0.21916343427456764	MSE: 949.9036346326164
Exponential Moving Averages (2016 Values) -	MAPE: 0.14830024461268773	MSE: 241.04551971326165

# 예측 회귀 모델(ML)

- Linear Regression
- Random Forest Regressor
- XgBoost Regressor

Error Metric Matrix (Tree Based Regression Methods) - MAPE

Linear Regression -	Train: 0.14746521303933213	Test: 0.13467526547991832
Random Forest Regression -	Train: 0.10218798838727976	Test: 0.1333350852029764
XgBoost Regression -	Train: 0.14341900306839336	Test: 0.13288520507863824

# 프로젝트를 진행하면서



1. 모빌리티 데이터에 대한 이해
2. 공간과 시간대별 수요량 측정
3. 새로운 모빌리티 서비스를 분석

# 모빌리티 서비스에 대한 이해

- 기본적으로 수요와 공급의 불균형의 문제, 정보 비대칭성의 문제를 지니고 있다.
- 사용자의 이동경로, 행위, 방문장소 등 데이터가 기존의 금융데이터, 검색데이터, **GPS**데이터, 대중교통 데이터의 비해 더 가치를 창출할 기회를 지녔다는 것을 알았다.
- 모빌리티 서비스는 이용자 자차를 소유여부, 차량 운행여부에 따라 서비스가 나뉜다.(자차 - 타인운전 => 대리, 타인 차 - 자신이 운전 => 카셰어링, 타인차 - 타인의 운전 => 헤일링 서비스)

## — 공간과 시간대별 수요량 측정

- 데이터마다 시각화를 통해 논리적으로 이해가지 않는 데이터들은 모두 아웃라이어로 제거하였다.
- 뉴욕시 안에서 클러스터링을 통해 비슷한 위치의 데이터들을 40개의 서브지역으로 분할할 수 있는 방법을 생각했다.
- 10분마다 시간을 분리하여 bin에 넣고, 그에 따라 수요량 책정이 가능한 방법을 이해했다.

# 새로운 모빌리티 서비스를 분석

- 현재 모빌리티 회사에서 가맹택시 서비스를 새로운 지역에서 론칭한다고 생각했을 때, 테스트 데이터를 통해 수요량을 측정한다면 의미가 있다고 생각한다.
- 모빌리티 리포트와 논문을 지속적으로 트래킹하면서 현재 소개된 딥러닝 모델들을 적용하여 성능을 좀 더 개선할 수 있는 방법을 추가해보는 것이 필요하다.



## 4. 마무리

→ 주요 인사이트

1. 모빌리티 서비스에 대한 이해

2. 공간과 시간대별 수요량 측정

3. 새로운 모빌리티 서비스를 분석