2024 DATA AI 분석 경진대회

대전 시내 택시 기사의 이동 위치 추천

팀 캡틴즈
이수민, 나영은, 이지현

[개요] 본 프로젝트는 대전 시내 택시 기사들의 운행 패턴을 분석하고, 택시 기사들의 수익을 극대화하기 위한 최적의 이동 위치를 추천하는 시스템을 개발하는 것을 목표로 한다. 2023년 4월부터 2024년 3월까지의 택시 영업 데이터를 바탕으로 승차 및 하차 위치, 시간, 요금, 주행 거리 등 기본 데이터를 활용하였으며, 추가적으로 기상청에서 제공한 기온 및 강수량 데이터와 대한민국 공휴일 정보, 대전광역시 택시 승강장 및 인기 관광지 정보를 결합해 분석을 진행하였다. 데이터 전처리 및 EDA를 통해 택시 수요 패턴을 파악한 후, KMeans 클러스터링 및 K-최근접 이웃(KNN) 알고리즘을 적용하여 요일별, 시간대별로 최적의 이동 경로를 추천하는 모델을 개발하였다. 이 시스템은 택시 기사들의 효율적인 운행을 돕고, 승객들이 택시를 더 쉽게 이용할 수 있는 환경을 제공한다.

1. 서 론

1. 문제 배경

대전은 대한민국의 주요 도시 중 하나로, 많은 인구가 도심과 외곽을 오가며 다양한 경제 활동을 펼치고 있다. 이러한 인구 밀집 지역에서 택시는 중요한 대중교통 수단 중 하나로, 시민들의 일상적인 이동과 물류 수송에 기여하고 있다. 그러나 최근 몇 년간 차량 공유 서비스 및 플랫폼의 등장으로 인해 전통적인 택시 산업은 큰 도전에 직면하고 있다. 많은 택시 기사들이 승객을 찾기 위해 장시간 대기하거나 비효율적으로 운행하게 되어 수익성 감소 문제를 겪고 있다.

더 나아가, 대도시의 교통 혼잡 문제는 점점 심화되고 있으며, 이로 인해 시민들의 이동 시간은 늘어나고, 환경적 부담도 증가하고 있다. 교통 혼잡으로 인해 에너지가 낭비되고 탄소 배출이 늘어나는 문제는 도시의 지속 가능한 발전을 저해하는 요인 중 하나이다. 택시 운행의 효율성 향상은 이러한 문제를 완화하는 데 중요한 역할을 할 수 있다.

특히 택시 기사들이 기존의 K 회사와 같은 연결 시스템에 의존해 승객을 확보하고 있으며, 이 과정에서 수익의 일부를 수수료로 지불하고 있는 현실은 많은 기사들에게 경제적 부담으로 작용하고 있다. 이에 따라 택시 기사들은 스스로 더 나은 수익을 창출할 수 있는 효율적인 운행 전략이 절실히 필요한 상황이다.

2. 목적

본 프로젝트는 이러한 문제를 해결하기 위해, 택시 기사들에게 특정 시간대와 요일에 승객을 확보할 가능성이 높은 이동 위치를 추천하는 시스템을 개발하는 것을 목표로 한다. 이를 통해 택시 기사들이 불필요한 공차 운행을 줄이고, 더 빠르게 승객을 찾을 수 있도록 도와주어 수익을 극대화할 수 있는 길을 열어준다. 또한 이 시스템은 교통 혼잡을 줄이고, 대기 시간을 단축하여 도시 내의 원활한 교통 흐름에도 긍정적인 영향을 미칠 것으로 기대된다. 더 나아가 이러한 변화는 택시 산업의 경쟁력 강화를 도모하고, 대전의 지속 가능한 교통 시스템 구축에도 기여할 것이다.

3. 활용 데이터

대전 시내 205대의 택시 영업 데이터를 활용한다. 이 데이터는 2023년 4월부터 2024년 3월까지 1년간의 영업 내역을 포함하고 있으며, 택시 기사들의 운행 패턴을 기반으로 승객 탑승 가능성이 높은 이동 위치를 추천하는 데 활용된다.

4. 추가 활용 데이터

종관기상관측(ASOS) (기상청 기상자료개방 포털) 데이터에서 강수, 기온 정보를 활용한다. 대전광역시 택시 승강장 현황 데이터(공공 데이터 포털)에서 132개의 택시승강장 정보를 활용하고 대전광역시 인기 관광지 현황 (한국관광 데이터랩) 데이터에서 상위 20개의 관광지 정보를 활용한다. 한국 공휴일 api (공공 데이터 포털)에서 2023.04~2024.03 기간의 공휴일 정보를 활용한다.

속성	설명	데이터 타입	참고
차량 이름	대전 시내 택시 차량 이름	String	차량 번호를 random으로 매핑
승차 시간	손님이 택시에 탑승한 시간	DateTime	'%Y-%m-%d %I:%M:%S %p' 혹은 '%Y-%m-%d %I:%M:%S'
승차 X 좌표 (위도)	탑승 위치의 위도	Float	-
승차 X 좌표 (경도)	탑승 위치의 경도	Float	-
요일	탑승 요일	DateTime	Monday-Sunday
하차 시간	손님이 택시에서 하차한 시간	DateTime	'%Y-%m-%d %I:%M:%S %p' 혹은 '%Y-%m-%d %I:%M:%S'
하차 X 좌표 (위도)	하차 위치의 위도	Float	-
하차 X 좌표 (경도)	하차 위치의 경도	Float	-

속성	설명	데이터 타입	참고
승차 거리	택시 탑승 거리	Float	m(미터)
할증 여부	택시 할증 구분	int	0(미할증), 1(할증), 2(복합할증)
요금	택시 요금	int	단위: 원

2. 본 론

1. 전처리 과정

전처리 과정을 통해 택시 이동 위치 추천 시스템의 정확도를 높일 수 있는 기초 데이터를 마련하였다.

1) 날짜 필터링 및 결측값 제거

먼저, 데이터셋에서 결측값을 포함하고 있는 인스턴스를 제거하였다. 변수에 결측값이 있을 경우 분석의 신뢰성을 떨어뜨릴 수 있다. 따라서 결측값을 가진 모든 인스턴스를 삭제하여 총 1156344개의 인스턴스에서 56652개의 유효한 인스턴스만을 남겼다.

다음으로, '승차시간' 컬럼을 datetime 형식으로 변환하고, 2023년과 2024년에 해당하지 않는 데이터는 모두 삭제한다. 프로젝트에서 다루는 기간이 2023년 4월부터 2024년 3월까지로 한정되었기 때문에, 이를 벗어나는 데이터를 제거하여 분석의 정확성을 높였고 데이터셋에서 '할증여부' 컬럼은 문자열로 저장되어 있었으며, '미할증'이라는 값은 할증이 적용되지 않았음을 의미한다. 이 값은 분석과 모델링에 적합하도록 정수형으로 변환되었으며, '미할증'이라는 값은 0으로 치환하였다.

승차시간'과 '하차시간'은 운행 시간 분석을 위해 매우 중요한 변수이다. 이 두 컬럼을 정확히 분석하기 위해 문자열로 저장된 시간 데이터를 datetime 형식으로 변환하였다.

승차시간'과 '하차시간'을 바탕으로 주행시간을 계산하여 새로운 '주행시간' 컬럼에 저장하고, 이 값을 보다 분석하기 용이하게 초 단위로 변환하여 '주행시간_초'라는 새로운 컬럼을 추가하였다. 558743개의 인스턴스에서 494790개의 인스턴스만 남겨 train_1.csv 에 저장한다.

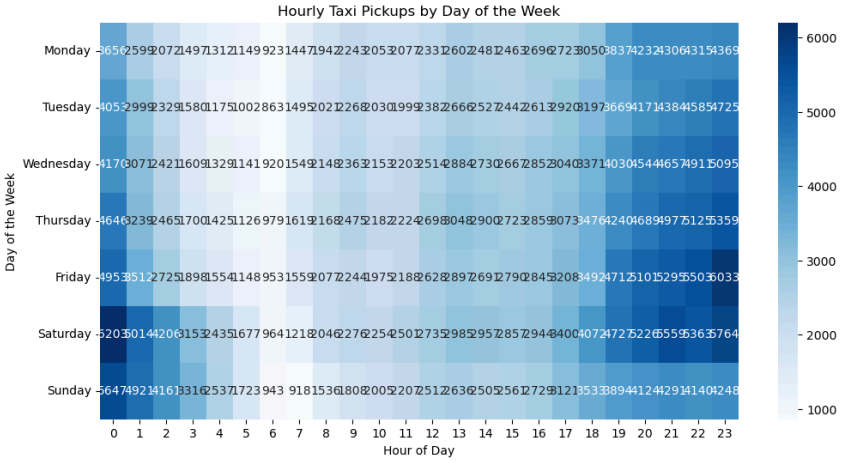
2) 기온, 강수량 데이터 추가 및 데이터 분할

날씨는 택시 수요에 큰 영향을 미치는 요인 중 하나이기 때문에, 분석에 기상 데이터 활용한다. 기상청 기상자료개방 포털의 종관기상관측(ASOS) 데이터를 활용하여, 택시 운행이 이루어진 날짜와 시간에 해당하는 기온 및 강수량 정보를 추가했다.

종관기상관측(ASOS) 데이터 결측값 처리 과정에서, 강수량 데이터의 결측값은 비가 내리지 않았다는 의미로 결측값을 '0'으로 대체하였다. 종관기상관측(ASOS) 데이터는 '일시' 컬럼에 날짜와 시간이

포함된 형태로 되어 있었기 때문에, 이를 택시 데이터와 결합하기 위해 연도, 월, 일, 시간 단위로 분할한다. 이를 통해 기상 데이터와 택시 운행 데이터의 시간 단위 일치를 확인하고, 두 데이터를 손쉽게 병합할 수 있도록 준비하였다. 이후 가공된 종관기상관측(ASOS) 데이터를 '대전_weather.csv' 파일로 저장하여 택시 운행 데이터와 병합했다.

택시 데이터의 '승차시간'을 datetime 형식으로 변환한 후, 이를 연도, 월, 일, 시간 단위로 분할하였다. 기온과 강수량 정보를 택시 데이터에 추가하기 위해, 두 데이터를 '연도', '월', '일', '시간'을 기준으로 병합하는 과정을 거쳤다. 택시 데이터의 '승차시간'과 기상 데이터의 해당 시간대 기온 및 강수량 정보를 매칭하여 결합한 후, 결측값을 제거하였다.



<표1> 시간 및 요일 별 택시 수요 시각화

시간대별, 요일별 택시 수요의 패턴을 분석하기 위해 택시 운행 데이터를 시각화한 결과, 특정 요일과 시간대에 따라 택시 수요가 명확히 다르게 나타나는 것을 확인할 수 있었다. 특히, 주중(월화수목)과 금요일, 그리고 주말(토일)의 택시 수요 패턴은 각기 다른 특성을 보였다.

- **주중(월화수목)**: 주중에는 출퇴근 시간대에 택시 수요가 급격히 증가하는 패턴을 확인할 수 있었다. 오전 7시에서 9시 사이, 그리고 저녁 6시에서 8시 사이에 수요가 가장 높게 나타나며, 이는 직장인들이 출퇴근 시간에 택시를 많이 이용하는 것으로 해석된다. 낮 시간대에도 꾸준한 수요가 이어지다가 저녁 시간이 되면서 다시 한번 수요가 상승하는 경향을 보였다.
- **금요일**: 금요일의 경우, 저녁 시간대에 택시 수요가 다른 요일에 비해 급격히 증가하는 패턴을 보였다. 이는 주말을 앞두고 외출 및 야간 활동이 증가하는 특성 때문으로 해석할 수 있다.
- **주말(토일):** 토요일과 일요일은 오전 시간대에는 비교적 수요가 낮은 편이었지만, 저녁 시간대로 갈수록 수요가 크게 증가하는 양상을 보였다.

택시 운행 데이터는 요일에 따라 승객 수요가 다르다는 것을 시각화를 통해 확인했고, 이에 데이터를 '주중', '금요일', '주말'로 분할하여 저장한다. 월요일부터 목요일까지는 'Weekdays'로 그룹화 후 'train_주중data1.csv'로 저장한다. 금요일은 'Friday'로 그룹화 후 'train_금요일data1.csv'로 저장한다. 토요일과 일요일은 'Weekend'로 그룹화 후 'train_주말data1.csv'로 저장한다.

이러한 차이를 반영하여 데이터를 분할하고, 각 그룹별로 클러스터링을 진행한다.. 이는 요일에 따른 택시 수요의 특성을 보다 명확히 분석하고, 그룹 내에서 유사한 패턴을 가진 클러스터를 식별하기 위함이다.

3) 공휴일 데이터 추가 및 최종 전처리 데이터 저장

택시 수요는 공휴일 여부에 따라 크게 달라질 수 있으므로, 공휴일 데이터를 추가하여 분석의 정밀도를 높였다. 공공데이터 포털에서 제공하는 한국 공휴일 API를 활용하여 2023년과 2024년의 공휴일 데이터를 수집하고 이를 택시 운행 데이터에 반영한다.
먼저, 공공데이터 포털 API를 이용해 2023년과 2024년의 공휴일 데이터를 수집하여 해당 연도의 공휴일 정보를 JSON 형식으로 받아와, 이를 CSV 파일로 저장하였다. (‘2023_Korean_holiday.csv', ‘2024_Korean_holiday.csv')
수집한 2023년과 2024년 공휴일 데이터를 불러와, 택시 운행 데이터에 '공휴일' 여부를 반영하였다. 공휴일 여부는 'year', 'month', 'day' 컬럼을 기준으로 택시 데이터와 매칭하였다.
각 데이터셋(주중, 금요일, 주말)에 대해 '공휴일' 여부를 확인하고, 공휴일인 경우 해당 열에 1을, 그렇지 않은 경우 0을 할당하였다. 주중, 금요일, 주말 데이터 각각에 대해 '공휴일' 정보를 추가하였다. 최종적으로 공휴일 정보를 포함한 주중, 금요일, 주말 데이터를 각각 최종 전처리 CSV 파일(‘train_주중data2.csv', ‘train_주말data2.csv', ‘train_금요일data2.csv')로 저장하였다. 이를 통해 공휴일 정보가 반영된 데이터를 기반으로 분석 및 모델링을 진행할 수 있게 되었다.

4) 대전광역시 택시 승강장 현황, 인기 관광지 현황 데이터 전처리

대전광역시 택시 승강장 현황 데이터(공공 데이터 포털) (‘인기관광지_20.csv’) 에서 132개의 택시승강장 정보를 활용하고 대전광역시 인기 관광지 현황 데이터(한국관광 데이터랩) (‘택시승강장.csv‘) 에서 상위 20개의 관광지 정보를 활용한다.
google maps 에서 api를 받아와 사물 주소를 위도, 경도로 변환하여 저장하고 기존 train 데이터에 추가하여 전처리한다. 전처리를 거쳐 ‘인기관광지_위도경도.csv’ , ‘택시 승강장_위도경도.csv’ 로 저장한다.

2. EDA

1) 대전광역시 택시 승강장 현황, 인기 관광지 현황 데이터 활용 시각화

택시 기사들에게 이동 위치를 추천하기 위해서는 대전광역시의 택시 승강장 위치와 주요 인기 관광지에 대한 정보가 중요하다. 이러한 정보를 시각화하여 패턴을 파악하기 위해, 대전 시내 택시 승강장 및 인기 관광지의 위치 데이터를 활용하였다.
월화수목 데이터, 금요일 데이터, 주말 데이터 각각을 시간대별로 나누어, 택시 승강장 및 인기 관광지 데이터를 지도 위에 시각화하였다. 이를 통해 시간대별로 어떤 위치에서 택시 승객을 쉽게 찾을 수 있는지 확인할 수 있으며, 특정 시간대에 인기 있는 승차 장소 및 관광지와의 관계를 시각적으로 분석할 수 있다.
Plotly의 지도 기능을 활용하여, 택시 승강장과 인기 관광지 위치를 지도 위에 표시하였다. 시간대별로 택시 승차 데이터를 추가하여, 특정 시간대에 어느 위치에서 택시 승객이 많이 발생하는지 파악할 수 있도록 했다. 지도는 슬라이더 기능을 추가하여 시간대별로 변화를 쉽게 확인할 수 있다. 지도에 시간대별 데이터를 반영하고, Plotly 슬라이더 기능을 통해 특정 시간대별로 지도 변화를 확인할 수 있도록 설정하였다. 이를 통해 사용자는 월요일부터 목요일까지 시간대별로 대전 시내에서 택시 승객 수요가 많아지는 지점들을 직관적으로 파악할 수 있다. 이 시각화를 통해 택시 기사들이 특정 시간대에 승객을 찾기 유리한 위치를 파악할 수 있으며, 택시 승강장과 관광지와의 상관관계도 확인할 수 있다. 이를 바탕으로 더 효율적인 운행 경로를 추천하는 데 기여할 수 있다.
* 시각화의 결과물은 ‘EDA_택시승강장인기관광지지도시각화.ipynb’ 참고.

2) 기사 성향 클러스터링

(1) 데이터 준비 및 전처리

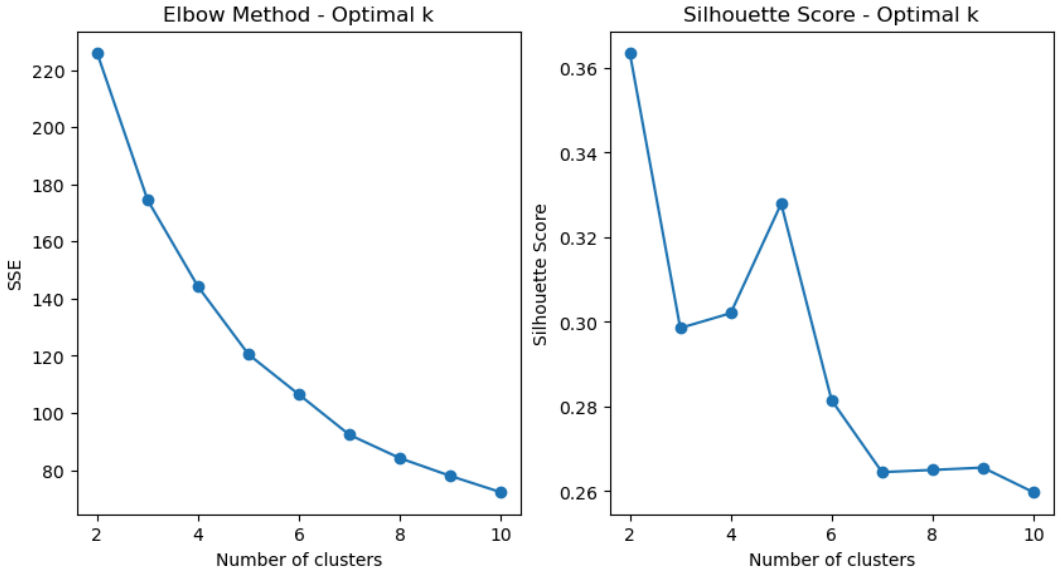
각 택시 운행에 대한 주행 시간(초 단위)을 계산하여 추가적인 특성을 생성하였고, '할증 여부'를 숫자형 데이터로 변환해 분석에 포함하였고, 데이터의 일관성을 유지하기 위해 '미할증'을 0으로 처리했다.

차량 이름을 기준으로 데이터를 그룹화하고, **평균 승차 거리, 평균 요금, 평균 할증 여부, 평균 주행 시간**을 계산하여 기사들의 운행 성향을 나타내는 새로운 집계 데이터를 생성하였다.

(2) 클러스터링 KMeans를 이용한 기사 성향 그룹화

전처리된 데이터를 바탕으로 KMeans 알고리즘을 사용해 택시 기사들의 성향을 그룹화했다. 각 택시 기사의 운행 패턴을 바탕으로 최적의 클러스터 개수를 찾기 위해 엘보우 방법과 실루엣 스코어를 사용하였다. 두 가지 방법을 통해 분석한 결과, **최적의 클러스터 개수는 3개**로 선정되었다.

- **엘보우 방법**: 클러스터 수에 따른 SSE(Sum of Squared Errors)를 확인한 결과, 클러스터 개수 3에서 SSE가 급격히 감소하는 지점이 나타났다.
- **실루엣 스코어**: 실루엣 점수를 확인한 결과, 클러스터 개수 3에서 가장 높은 값을 보였다.



<표2> 기사 성향 엘보우 방법 및 실루엣 스코어

이를 바탕으로 KMeans 알고리즘을 적용해 3개의 클러스터로 택시 기사들을 그룹화했다. 각 클러스터는 승차 거리, 요금, 주행 시간, 할증 여부 등의 특성에서 서로 다른 경향을 보였다.

(3) 기사 성향 클러스터링 결과

각 클러스터는 **승차 거리**, **요금**, **할증 여부**, 주행 시간(초 단위)에서 다른 경향을 보였다.

[Cluster 0]

- 평균 승차 거리가 가장 길며(5689m), 요금도 가장 높다(8493원).
- 할증 여부의 평균은 0.124로, 할증 운행 비율이 다른 클러스터보다 높다.
- 주행 시간은 평균 747.8초로, 클러스터 중 가장 긴 운행 시간이 특징이다.

→ 장거리 운행 및 비교적 고요금 지역에서의 운행 패턴을 반영한다.

[Cluster 1]

- 평균 승차 거리는 가장 짧고(4934m), 요금도 가장 낮다(7631원).
- 할증 여부는 0.057로, 할증 운행이 상대적으로 적게 발생한다.
- 주행 시간은 720.99초로, 클러스터 0보다 약간 짧다.

→ 주로 단거리 운행을 선호하는 기사들이 속하는 클러스터이다.

[Cluster 2]
- 평균 승차 거리는 5214.5m, 요금은 8018.6원으로 중간값을 보인다.
- 할증 여부는 0.358로, 할증 운행 비율이 매우 높아 이를 선호하는 기사들의 특징을 반영한다.
- 주행 시간은 627.88초로, 비교적 짧은 주행 시간이 특징이다.

→ 할증 운행을 자주 수행하면서도 비교적 짧은 거리에서 운행을 마치는 경향이 나타났다.

(4) 변수 중요도 분석 (Random Forest)

KMeans로 분류된 각 클러스터에 대해 변수 중요도를 평가하기 위해 랜덤 포레스트(Random Forest) 알고리즘을 사용했다. 이 과정을 통해 택시 기사의 운행 성향을 결정짓는 주요 변수들을 분석했다.

분석 결과, 할증 여부와 주행 시간이 기사 성향을 구분하는 데 가장 중요한 변수로 나타났으며, 이 두 변수가 클러스터 구분에 큰 영향을 미쳤다.

변수 중요도는 할증여부 0.382649, 요금 0.286235, 승차거리(m)0.183617, 주행시간_초 0.147499 순으로 나타났다. 이 분석을 통해, 각 택시 기사의 성향을 파악하고 클러스터별 운행 패턴을 보다 명확하게 구분할 수 있었다.

3. 클러스터링

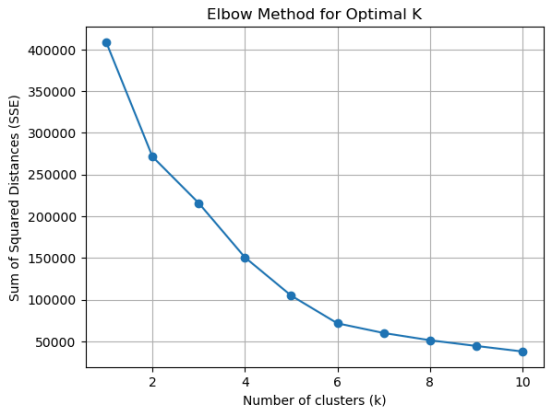
1) 최적의 클러스터 수(K) 결정 – 엘보우 및 실루엣 분석

클러스터링을 적용하기에 앞서, 최적의 클러스터 수를 결정하기 위해 두 가지 방법을 사용했다. : 엘보우 방법(Elbow Method)과 실루엣 스코어(Silhouette Score).

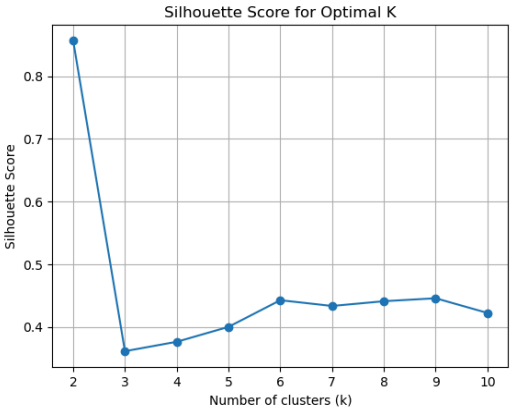
(1) **엘보우 방법의 결과 해석**: 엘보우 방법에서는 k=3 또는 k=4에서 SSE의 감소가 완만해지는 구간을 볼 수 있다. 이를 바탕으로 이 지점에서 클러스터 개수를 선택하는 것이 합리적이다. 따라서 엘보우 방법만을 고려한다면 k=3 또는 k=4가 적합하다.

(2) **실루엣 점수 해석:** 실루엣 점수는 클러스터링의 품질을 나타내는 지표로, 점수가 높을수록 클러스터 내부의 응집력이 높고 클러스터 간의 분리가 잘 이루어진다는 것을 의미한다. 실루엣 점수가 가장 높은 지점이 최적의 클러스터 개수로 해석된다. 그래프에서는 k=2에서 가장 높은 실루엣 점수(약 0.8)를 얻고 있지만, k=3 또는 k=4에서도 어느 정도 실루엣 점수가 유지되고 있다.

(3) **결론:** k=2는 실루엣 점수 측면에서 최적의 클러스터 개수로 보이지만, 2개의 클러스터는 데이터의 패턴을 지나치게 단순화할 가능성이 있다. k=3 또는 k=4는 엘보우 방법에서 추천된 값이며, 실루엣 점수도 0.6~0.7로 여전히 준수한 수준을 유지하고 있다. 따라서 k=3을 선택하는 것이 균형적인 선택일 수 있다. 이 값은 엘보우 방법에서도 추천되었고, 실루엣 점수도 여전히 준수한 수준을 유지하므로 클러스터링의 품질과 데이터의 복잡성을 적절히 반영할 수 있다.

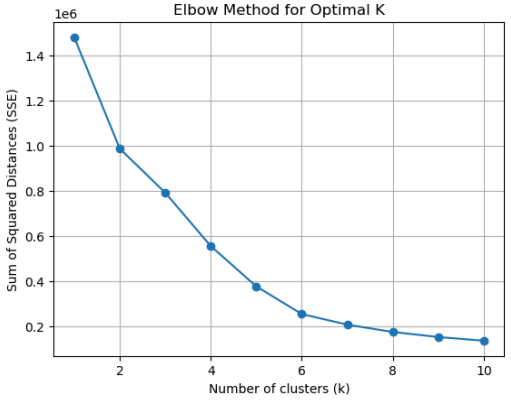


<표3> 금요일 최적의 클러스터 수 결정 -엘보우

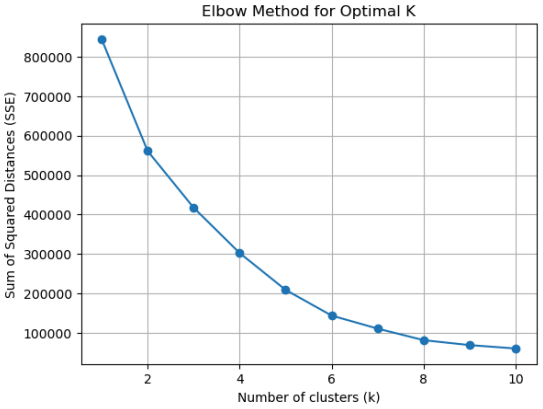


<표4> 금요일 최적의 클러스터 수 결정 -실루엣 스코어

주말, 주중 데이터도 이러한 방식으로 최적의 클러스터 수(K= 3)를 도출해냈다.



<표5> 주중 최적의 클러스터 수 결정 -엘보우



<표6> 주말 최적의 클러스터 수 결정 -엘보우

2) 클러스터링 적용

최적의 클러스터 수(K=3)를 선정한 후, KMeans 알고리즘을 사용하여 클러스터링을 진행하였다. 이 과정에서 사용된 주요 특성은 **승차 위치(위도/경도), 시간(hour), 기온(°C), 강수량(mm)**, **공휴일**이다. 데이터를 클러스터링을 수행하기 전에 각 특성의 범위를 맞추기 위해 표준화를 적용하고, 승차 좌표가 결측되거나 0인 데이터를 제거하였다.

KMeans는 계산이 빠르고, 클러스터가 명확한 중심을 가지며, 다양한 특성의 데이터를 직관적으로 처리할 수 있어 이 프로젝트에 적합한 알고리즘이라고 판단했다.

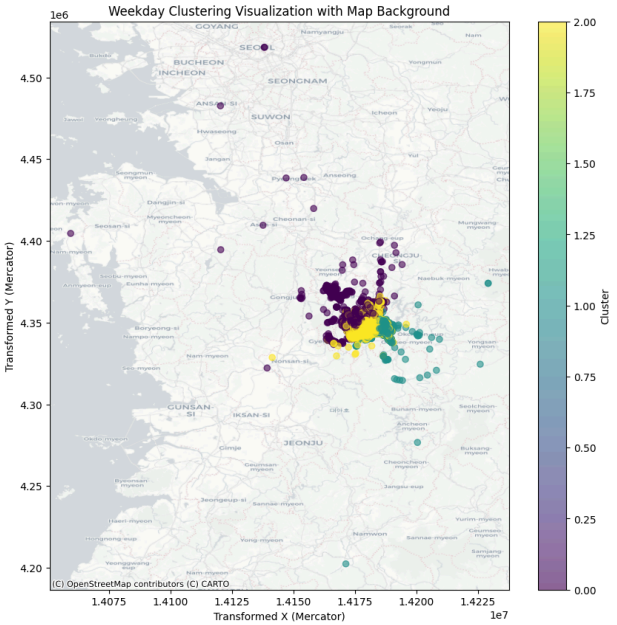
DBSCAN은 노이즈 처리에 강하지만, 택시 수요 데이터의 클러스터 밀도가 일정하지 않아서 클러스터 형성이 불균일하게 될 수 있고, eps와 min_samples 파라미터의 최적값을 찾기 어려워 실무 적용에 한계가 있다고 판단했다. 또, **Hierarchical Clustering**은 데이터가 많을수록 계산 비용이 매우 높아지며, 클러스터 구조가 고정되어 있어 유연성이 부족해 우리 모델에 부적합하다고 판단했다.

3) 데이터셋 별 클러스터링 시각화

각 데이터셋(주중, 금요일, 주말)에 대해 k=3으로 k-means 클러스터링을 적용하고,

(1) 주중 데이터

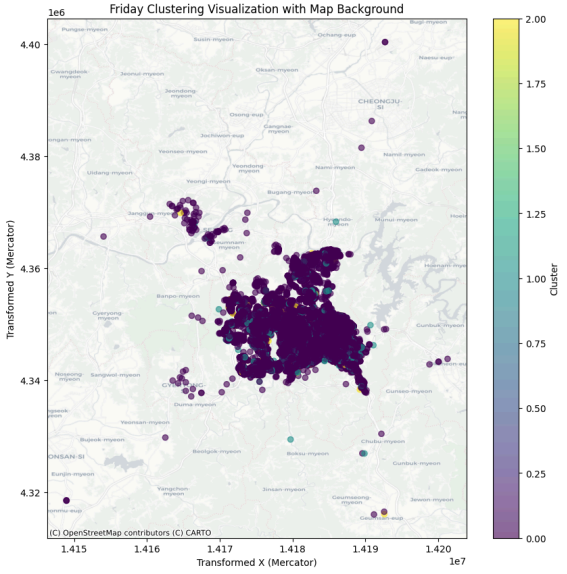
시각화 결과 세 클러스터가 지리적으로 다른 분포를 보이는 것을 확인할 수 있다. 클러스터 0의 데이터들은 대전광역시의 북서쪽(유성구)에 많이 분포하며, 세종시와 청주시의 승차 데이터도 확인할 수 있다. 클러스터 1의 데이터들은 대부분 대전광역시의 중심부(대덕구, 서구, 중구)에 분포한다. 클러스터 2의 데이터들은 대전광역시의 남동쪽(동구)에 많이 분포하며, 옥천군과 금산군에도 일부 분포한다.



<표7> 주중 데이터 클러스터링 시각화

(2) 금요일 데이터

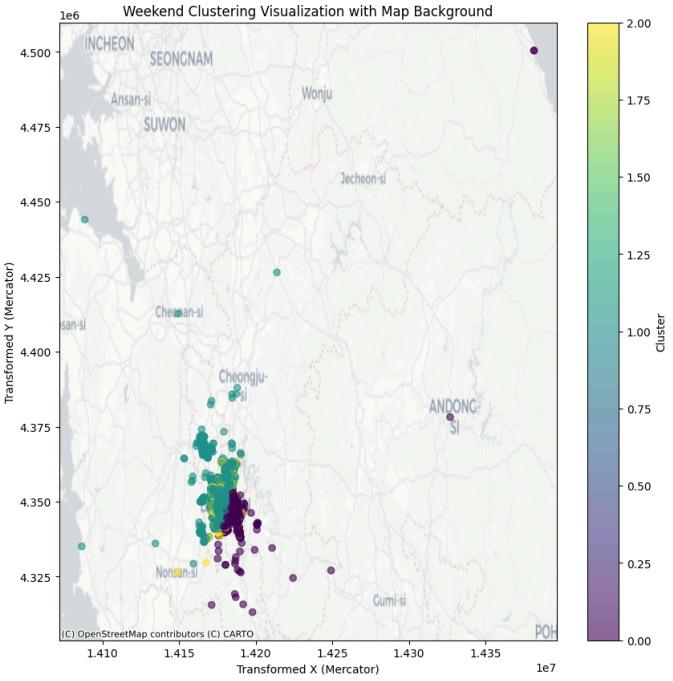
시각화 결과 세 클러스터가 지리적으로 비슷한 분포를 보이는 것을 확인할 수 있다. 금요일 데이터에서 대전광역시를 크게 벗어나는 데이터포인트는 적은 편이며, 세 클러스터 모두 대전 광역시 내에서 고른 분포를 보이는 것을 알 수 있다. 클러스터 0의 경우 세종특별시에서도 일부 데이터 포인트들이 밀집되어 있는 것을 확인할 수 있었다.



<표8> 금요일 데이터 클러스터링 시각화

(3) 주말 데이터 (토일)

시각화 결과 세 클러스터의 지리적 분포에 차이를 확인할 수 있었다. 클러스터 0은 대전광역시의 남동쪽(동구, 중구)에 주로 분포해있으며, 대전광역시를 벗어나는 데이터포인트들도 일부 존재했다. 클러스터 1은 대전광역시의 북서쪽(유성구, 대덕구, 서구)에 주로 분포했으며, 세종특별시에서도 일부 데이터포인트들이 존재하는 것을 볼 수 있다. 클러스터 2는 전반적으로 대전광역시 내에서 고르게 분포해 있으며, 세 클러스터 중 대전 밖의 데이터 포인트의 개수가 가장 적었다.



<표9> 주말 데이터 클러스터링 시각화

4) 클러스터링 결과 분석

요일별로 클러스터링된 데이터를 바탕으로 각 클러스터의 시간(hour), 기온(°C), 강수량(mm)의 특성을 분석하였다. 3개의 클러스터 각각에 대해 분석한 내용은 다음과 같다.

(1) 주중 데이터 클러스터링 결과 분석

	cluster_multi	승차X좌표		승차Y좌표		hour		기온(°C)		강수량(mm)		holiday
		mean	std	mean	std	mean	std	mean	std	mean	std	mean
0	0	127.388469	0.037337	36.350260	0.030625	5.832985	4.273152	10.619039	9.687180	0.159094	1.172381	0.0
1	1	127.393513	0.037316	36.346031	0.027949	18.013446	3.653437	17.073030	9.943828	0.388855	2.176843	0.0
2	2	127.393432	0.037625	36.346162	0.027726	12.942510	7.673655	13.814939	10.123347	0.000000	0.000000	1.0

<표 10> 주중 데이터 클러스터링 결과

[Cluster 0]

- 시간대(hour): 평균 5.83시로, 주로 아침 시간대에 택시 수요가 집중된다.
- 기온(°C): 평균 기온은 10.62°C로, 세 클러스터 중 가장 낮다.
- 강수량(mm): 평균 강수량은 0.16mm로, 비가 거의 오지 않는 날이 많았다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

[Cluster 1]
- 시간대(hour): 평균 18.01시(오후 6시)로, 저녁 시간대에 수요가 많다.
- 기온(°C): 평균 기온 17.07°C로, 세 클러스터 중 가장 높다.
- 강수량(mm): 평균 강수량은 0.38mm로, 비가 내리는 날이 많았다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

[Cluster 2]
- 시간대(hour): 평균 12.94시(오후 1시)로, 정오 즈음에 수요가 집중된다.
- 기온(°C): 평균 기온 13.81°C로, 쌀쌀한 날씨에서 수요가 높다.
- 강수량(mm): 평균 강수량은 0.00mm로, 모든 데이터에서 강수량이 0이다.
- holiday: 평균 1.0으로, 모든 데이터가 공휴일에 해당한다.

이 분석 결과, 주중 클러스터들은 시간대별로 명확한 차이를 보이며, 강수량에 따라서도 수요가 달라지는 경향이 확인되었다. 표준편차를 확인한 결과 각 클러스터 내에서 기온은 비교적 넓은 분포를 보였다.

(2) 금요일 데이터 클러스터링 결과 분석

cluster_multi	승차X좌표		승차Y좌표		hour		기온(°C)		강수량(mm)		holiday
	mean	std	mean	std	mean	std	mean	std	mean	std	mean
0	0	127.395351	0.037024	36.344414	0.026460	18.212280	3.593482	16.530340	10.086332	0.359801	2.107659	0.0
1	1	127.386026	0.036814	36.352481	0.030464	5.498846	4.232761	11.705104	9.546715	0.258058	1.599133	0.0
2	2	127.393055	0.037250	36.344336	0.025071	12.665478	7.705710	6.690894	9.540458	1.357846	2.664730	1.0

<표 11> 주중 데이터 클러스터링 결과

[Cluster 0]
- 시간대(hour): 평균 18.21시(오후 6시)로, 저녁 시간대에 수요가 집중되었다.
- 기온(°C): 평균 16.53°C로, 세 클러스터 중 가장 높았다.
- 강수량(mm): 평균 0.35mm로, 비가 내리는 날이 많았다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

[Cluster 1]
- 시간대(hour): 평균 5.50시로, 새벽 시간에 수요가 집중된다.
- 기온(°C): 평균 기온 11.71°C로, 비교적 쌀쌀한 날이 많았다.
- 강수량(mm): 평균 강수량 0.26mm로, 비가 약간 내리는 날이 많았다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

[Cluster 2]
- 시간대(hour): 평균 12.67시로, 정오 시간대에 수요가 집중되었다.
- 기온(°C): 평균 기온 6.69°C로, 기온이 낮은 날이 많았다.
- 강수량(mm): 평균 강수량 1.36mm로, 세 클러스터 중 가장 높았다.
- holiday: 평균 1.0으로, 모든 데이터가 공휴일에 해당한다.

이 분석 결과, 주중 클러스터들은 시간대별로 명확한 차이를 보이며, 강수량은 클러스터 0, 1에서 비슷한 수준으로 확인되었다. 표준편차를 확인한 결과 각 클러스터 내에서 기온은 비교적 넓은 분포를 보였으나, 클러스터 간에는 유의미한 차이가 있었다.

(3) 주말 데이터 클러스터링 결과 분석

	cluster_multi	승차X좌표		승차Y좌표		hour		기온(°C)		강수량(mm)		holiday
		mean	std	mean	std	mean	std	mean	std	mean	std	mean
0	0	127.388042	0.037275	36.347683	0.027749	4.470991	3.921329	10.115986	9.508570	0.054107	0.419740	0.0
1	1	127.395441	0.038602	36.344814	0.026992	17.753244	3.845039	16.825508	10.060002	0.191918	1.295509	0.0
2	2	127.396324	0.037206	36.344470	0.026330	13.307087	7.657154	1.479318	2.991980	0.000000	0.000000	1.0

<표 12> 주말 데이터 클러스터링 결과

[Cluster 0]
- 시간대(hour): 평균 4.47시로, 새벽 시간대에 주로 택시 수요가 발생한다.
- 기온(°C): 평균 기온 10.11°C로, 비교적 쌀쌀한 날이 많았다.
- 강수량(mm): 평균 강수량 0.05mm로, 비가 거의 오지 않는 날이 많았다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

[Cluster 1]
- 시간대(hour): 평균 17.75시(오후 6시)로, 저녁 시간대에 수요가 집중된다.
- 기온(°C): 평균 기온 16.83°C로, 따뜻한 날씨에서 주로 수요가 발생한다.
- 강수량(mm): 평균 강수량 0.19mm로, 세 클러스터 중 가장 높다.
- holiday: 평균 0.0으로, 공휴일에 해당하는 데이터가 없었다.

.

[Cluster 2]
- 시간대(hour): 평균 13.31시(오후 1시)로, 오후 시간대가 많았다.
- 기온(°C): 평균 기온 1.48°C로, 기온이 매우 낮은 날이 많았다.
- 강수량(mm): 평균 강수량 0.00mm로, 비가 내린 날이 없었다.
- holiday: 평균 1.0으로, 모든 데이터가 공휴일에 해당한다.

이 분석 결과, 주말 클러스터들은 주로 저녁 시간대에 택시 수요가 집중되는 경향이 있었다. Cluster 0과 Cluster 1은 저녁 시간대의 따뜻한 날씨에 주로 수요가 발생하는 패턴을 보였으며, Cluster 2는 이른 새벽 시간대의 쌀쌀한 날씨에도 수요가 나타났다. Cluster 4는 정오 시간대에 따뜻한 날씨에서 택시 수요가 증가하는 경향을 보였으며, 비가 많이 오는 날에도 수요가 꾸준히 발생했다.

종합적으로 세 그룹(주중, 금요일, 주말)에서 모두 공휴일 데이터들만 존재하는 클러스터가 생성되는 것을 볼 수 있는데, 이를 나머지 2개 클러스터와 비교해보면 'hour' 속성에서 공휴일 클러스터의 표준편차가 유의미하게 큰 것이 확인된다. 이는 공휴일에 해당하는 데이터의 클러스터가 시간대별 택시 수요의 변동이 매우 크다는 점을 시사한다. 공휴일에 다양한 요인(예: 행사, 여행 등)이 복합적으로 작용하여 특정 시간대에 택시 수요가 집중되지 않고, 보다 넓은 범위에서 분산될 가능성이 있다. 반면, 비공휴일 클러스터에서는 특정 시간대(예: 출퇴근 시간대 또는 저녁 시간대)에 택시 수요가 집중되는 경향이 더 뚜렷하게 나타났다. 이러한 분석을 바탕으로, 공휴일에는 시간대별로 더욱 유연한 택시 배차

전략이 필요하며, 반대로 평일이나 주말에는 주로 수요가 높은 시간대에 맞춰 택시 운행을 최적화할 수 있는 전략이 요구된다.

4. 모델 개발

모델 개발 과정에서는 택시 기사의 위치, 시간, 기온, 강수량, 공휴일 등 입력 데이터를 기반으로 최적의 이동 위치를 추천하는 시스템을 구축하였다. 이 시스템은 요일별로 학습된 클러스터링 모델을 사용하여 현재 상황에 맞는 클러스터를 선택하고, 해당 클러스터 내에서 빈번하게 발생한 위치를 추천하는 방식으로 작동한다. 주요 단계는 다음과 같다.

1) 요일별 모델 선택

본 프로젝트는 택시 수요가 요일별로 상이하게 나타나는 특징에 주목하여, 요일에 따라 개별 클러스터링 모델을 적용하는 방안을 도입하였다. 이를 위해 choose_model_by_day 함수를 통해 입력된 요일(day)에 따라 주중(월목), 금요일, 주말(토일)로 구분하여 각각의 요일 군에 적합한 클러스터링 모델을 선택하도록 설계하였다. 이를 통해 주중, 금요일, 주말의 수요 패턴을 반영한 분석이 가능해졌으며, 택시 기사가 보다 수익을 극대화할 수 있는 이동 위치 추천을 제공할 수 있게 되었다.

2) 클러스터 할당을 통한 택시 수요 예측

특정 위치, 시간, 기온, 강수량, 공휴일 여부에 따른 택시 수요의 차이를 반영하기 위해 get_cluster_for_input 함수를 설계하였다. 이 함수는 입력된 정보(좌표, 시간, 기온, 강수량, 공휴일 여부)를 바탕으로, 클러스터링 모델을 이용해 가장 적합한 클러스터를 예측하여 할당하는 기능을 수행한다.

먼저, 클러스터링에 사용된 특성(좌표, 시간, 기온, 강수량, 공휴일 여부)을 기준으로 입력 데이터와 기존 데이터셋을 표준화하였다. StandardScaler를 사용해 각 특성을 표준화함으로써, 데이터의 분포에 맞춰 동일한 범위로 조정하였다.

다음으로, KMeans 클러스터링 모델 생성한다. 클러스터 개수(n_clusters)는 기존 데이터의 클러스터 개수를 기반으로 설정하였으며, KMeans 알고리즘을 적용하여 클러스터링을 수행하였다. 이로써 택시 수요 패턴이 유사한 지역과 시간대를 효과적으로 군집화하였다.

마지막으로, 클러스터 할당한다. 입력된 좌표, 시간 등 특성 값을 표준화한 후 클러스터 모델을 통해 해당 데이터의 클러스터 레이블을 예측하였다. 예측된 클러스터는 특정 상황에서 택시 수요가 높을 가능성이 있는 위치를 파악하는 데 활용된다. 이러한 방법을 통해 택시 기사는 자신이 있는 위치와 현재 상황에 맞는 클러스터를 예측할 수 있으며, 이를 바탕으로 적절한 이동 방향을 결정할 수 있다.

3) 최적 이동 위치 추천

recommend_top_locations 함수는 예측된 클러스터 내에서 가장 높은 빈도로 승객이 탑승했던 위치를 파악하여, 수익을 극대화할 수 있는 이동 위치를 추천하는 기능을 수행한다. 클러스터 내 승객 승차 좌표를 빈도 기준으로 정렬하여 상위 15개의 위치를 추천하는 방식으로, 특정 요일과 시간대에 최적의 위치를 제공할 수 있도록 설계하였다.

이와 같은 시스템은 특정 클러스터 내에서 가장 수요가 높은 장소를 효율적으로 식별하고, 택시 기사가 빈번한 승객 탑승 지점으로 이동하도록 돕는다. 이를 통해 택시 기사는 수익을 극대화할 수 있으며, 동시에 승객들에게 더 나은 접근성을 제공할 수 있다.

4) 관광지 및 택시 승강장을 고려한 최적 위치 추천

본 시스템에서는 택시 기사에게 수익을 극대화할 수 있는 위치를 보다 정교하게 추천하기 위해, 빈도 동점인 경우 관광지와 택시 승강장까지의 거리를 활용해 재정렬하는 기능을 추가하였다. 관광지와 택시 승강장 주변에는 택시 수요가 집중될 가능성이 높기 때문에, 해당 위치와의 거리를 가중치로 적용해 순위를 정함으로써 보다 수요에 부합하는 추천 위치를 제공할 수 있다.

5) 클러스터 중심과의 거리 계산

haversine_distance 함수는 클러스터 중심과 관광지, 택시 승강장 간 거리를 계산하는 데 사용되었다. 이를 통해 클러스터 내에서 수요가 높은 상위 좌표(top locations)가 빈도가 동일할 경우, 해당 좌표에서 가까운 관광지나 승강장과의 거리 차이를 반영하여 우선 순위를 재정렬하였다.

6) 빈도 동점 처리 및 가중치 적용

rank_by_weighted_distance 함수는 특정 클러스터 내 상위 좌표들의 빈도가 동일한 경우, 관광지 및 택시 승강장과의 거리를 반영하여 우선 순위를 정한다. 해당 좌표와 가장 가까운 관광지까지의 최소 거리를 계산하고, 해당 좌표와 가장 가까운 택시 승강장까지의 최소 거리를 계산한다. 관광지와 택시 승강장 거리의 가중치를 각각 0.7과 0.3으로 적용하여 최종 점수를 산출한다. 관광지와의 거리에 더 높은 가중치를 부여함으로써, 관광지 주변에 수요가 집중되는 특성을 반영하였다. 빈도가 높은 순으로 정렬하되, 동점인 경우 가중치를 적용한 최종 점수를 기준으로 정렬하였다.

7) 관광지 및 승강장 반영 위치 추천

recommend_locations_with_weights 함수는 요일, 시간, 날씨, 공휴일 여부 등을 기반으로 클러스터를 선택하고, 해당 클러스터 내에서 관광지 및 택시 승강장 정보를 반영한 위치 추천을 수행한다. 클러스터 내 빈도가 높은 좌표 중에서 관광지나 승강장과 가까운 순서로 재정렬하여, 최종적으로 상위 10개의 위치를 추천한다. 이러한 접근은 특히 관광지 및 주요 지점에 대한 택시 수요가 높아지는 시간대나 주말, 공휴일에 택시 기사들이 효과적으로 수익을 극대화하는 데 도움을 줄 수 있다.

위와 같은 방식으로 빈도 및 관광지, 승강장과의 거리를 고려한 위치 추천 시스템은 택시 수요가 집중되는 위치를 정교하게 파악하고, 기사들이 최적의 위치로 이동할 수 있도록 돕는다.

5. 실험 및 평가

1) 검증 데이터 셋 준비

본 팀이 개발한 알고리즘의 성능을 검증하기 위해 검증 데이터셋을 준비해야 한다. 실제 택시의 영업장소는 24시간 * 7일을 기준으로 대전 시내 임시 위치 20곳의 위치를 선별하여 3360 case를 선별하여 성능 검증을 진행하여야 한다. 본 팀은 대전 시내 내에서 20곳의 위치를 선정하였고 그 장소는 이러하다.

대전역 인근(동구) – (36.351823, 127.385284)	대동 인근 (동구) – (36.337942, 127.374628
대전천 주변(중구) – (36.338762, 127.396842)	갑천 주변 (서구) – (36.359872, 127.387291)

용전동, 대전복합터미널 근처(동구) - (36.325961, 127.409374)	대전역 동쪽, 자양동 근처 (동구) - (36.333528, 127.409182)
은행동 으능정이거리 주변(중구) - (36.344561, 127.378295)	서대전 네거리 인근 (중구) - (36.347092, 127.386542)
대전광역시청 근처(서구) - (36.355489, 127.399884)	태평동 부근 (중구) - (36.340875, 127.379856)
충남대학교 대덕캠퍼스 근처(유성구) - (36.331275, 127.392154)	대전 갤러리아백화점 타임월드 근처 (서구) - (36.352761, 127.404273)
정부대전청사 주변(서구) - (36.360042, 127.394687)	판암동 인근 (동구) - (36.328694, 127.390572)
대전복합터미널 근처(동구) - (36.327489, 127.405289)	용두동 주변 (중구) - (36.341982, 127.392874)
대흥동 주변 (중구) - (36.342873, 127.382746)	유천동 근처 (중구) - (36.355481, 127.380293)
둔산동 일대 (서구)- (36.349561, 127.398925)	용문동 인근 (서구) - (36.334521, 127.401983)

또한 7일간의 데이터는 제공된 테스트 데이터에서 임의로 선택한 날짜들로 구성하였다. 이 날짜들은 2023년 5월 29일(월요일), 2023년 7월 29일(토요일), 2023년 9월 29일(금요일), 2023년 11월 26일(일요일), 2023년 11월 29일(수요일), 2024년 1월 29일(월요일), 2024년 3월 29일(금요일)이다.

즉, 검증을 진행할 case는 20장소 * 7일 * 24시간 = 3360 case이다. 3360 case의 검증데이터를 만들기 위해, 대전_weather.csv 파일을 불러와서 선정한 7개의 날짜를 추출하고, 각 날짜에 해당하는 요일과 공휴일 여부를 매칭시킨 뒤 필요한 열만 추출하여 24시간 * 7일에 대한 20개의 장소를 부여해 'valid_data.csv'로 저장하여 최종 검증 데이터셋을 만든다. 이렇게 생성된 검증 데이터셋에 대해 각각 10개의 추천 위치(총 33600행)를 도출하고, 추천된 10개의 위치가 테스트 데이터셋의 해당 요일 그룹(주중, 금요일, 주말)의 시간대에 택시 승객이 있는 경우 정답으로 간주하여 precision-recall을 계산할 것이다.

2) 정답률 확인을 위한 test 데이터 준비

33600개의 검증데이터와 test 데이터(제공된 데이터)와의 일치도를 확인하기 위해 test 데이터는 전처리 과정이 필요하다. 일단, test_taxi_tims_U.csv 파일을 불러와서 필요한 열인 승차시간, 승차요일,

승차x좌표, 승차y좌표를 뽑고, 검증에 사용하는 7개의 날짜를 추출해온다. 그 다음 요일별로 그룹(주중, 금요일, 주말)을 생성하여 그룹별 테스트 데이터의 개수를 확인한다. weekdays(주중)은 5004개, weekend(주말)은 3557개, Friday(금요일)은 2959개인 것을 확인하였고, 데이터의 최소 개수인 2959개로 모든 그룹의 개수를 맞춰준다. 그 후 승차 x좌표와 y좌표를 train데이터로 추천 작업을 실행할 때 반올림하여 진행하였으므로, test 데이터에도 소수점 넷째자리까지 반올림하여 자릿수를 맞춰준 뒤 balanced_data.csv 파일로 저장한다.

3) 검증 실험 및 정답률 계산

검증을 진행하기 위해 weekday_clusterd, friday_clustered, weekend_clustered 클러스터링 데이터를 불러오고, 앞서 만든 추천 모델(함수)을 적용한다. input_data로는 valid_data.csv 파일을 불러와 사용하며, 이 파일의 요일, 경도, 위도, 시간대, 기온, 강수량, 공휴일 여부를 입력값으로 받아 관광지 및 택시 승강장을 고려한 추천 위치 10개를 3360개의 케이스에 대해 산출한다. 이렇게 생성된 33600개 행의 데이터프레임은 날짜, 요일, (현재)위도, (현재)경도, 기온, 강수량, 추천순위, 추천 위도, 추천 경도, 빈도수, 가중치점수, 클러스터 열을 포함한 recommendation_df로 저장하고, 이 데이터프레임과 balanced_data를 비교하여 정답률을 확인한다.

정답률을 산출하는 방법은 다음과 같다. 예를 들어, 2023년 5월 29일 월요일 0시의 추천 좌표 10개의 정답률을 확인하려면, balanced_data(전처리된 테스트 데이터)에서 월요일에 해당하는 주중 그룹의 0시 승차 좌표들 중 추천된 10개 좌표가 몇 개나 포함되는지를 확인하고, 포함된 개수에 따라 백분율로 표시한다. 이 과정의 구현은 calculate_accuracy_percentage 함수에서 진행하였다. 이렇게 3360 케이스에 대한 추천 좌표의 최종 정답률을 계산한 뒤, result_ac.csv 파일로 저장한다.

4) 결과 해석

3360 case에 대해 정답률을 계산한 결과, 70%의 정답률을 가진 case는 40개, 60%는 240개, 50%는 360개, 40%는 508개, 30%는 602개, 20%는 538개, 10%는 532개, 0%는 540개의 결과가 나왔다. 그 중 대표적인 몇 가지의 case를 갖고 해석을 진행하려고 한다.
먼저, 정답률이 70%인 데이터를 살펴보면, 2023년 7월 29일 토요일 새벽 2시의 경우 정답률이 70%로 나왔으며 클러스터가 0으로 배정되었음을 확인할 수 있다. 이는 주말 그룹의 클러스터 0 설명인 "주말 쌀쌀한 새벽 시간대에 승차가 많은 곳 리스트"와 일치하는 결과로, 해당 시간대에 높은 정답률로 추천된 점에서 추천 모델의 설득력을 확인할 수 있다.

	날짜	요일	위도	경도	클러스터	정답률 퍼센트
0	2023-07-29 02:00:00	Saturday	36.351823	127.385284	0	70.0
1	2023-07-29 02:00:00	Saturday	36.359872	127.387291	0	70.0
2	2023-07-29 02:00:00	Saturday	36.340875	127.379856	0	70.0
3	2023-07-29 02:00:00	Saturday	36.337942	127.374628	0	70.0
4	2023-07-29 02:00:00	Saturday	36.334521	127.401983	0	70.0
5	2023-07-29 02:00:00	Saturday	36.349561	127.398925	0	70.0

<표 13> 결과 해석 – 2023년 7월 29일 토요일 새벽 2시

다음으로, 정답률이 60%인 데이터 중 2024년 3월 29일 금요일 23시의 경우를 보면, 정답률이 60%로 나타났으며 클러스터가 0으로 배정되었음을 확인할 수 있다. 이는 금요일 그룹의 클러스터 0 설명인

"금요일 저녁 흐린 시간대에 승차가 많은 곳 리스트"와 부합하는 결과로, 추천이 적절하게 이루어졌다고 해석할 수 있다.

236	2024-03-29 23:00:00	Friday	36.340875	127.379856	0	60.0
237	2024-03-29 23:00:00	Friday	36.352761	127.404273	0	60.0
238	2024-03-29 23:00:00	Friday	36.341982	127.392874	0	60.0
239	2024-03-29 23:00:00	Friday	36.334521	127.401983	0	60.0

<표 14> 결과 해석 – 2023년 3월 29일 금요일 23시

마지막으로, 정답률이 50%인 데이터 중 2023년 5월 29일 월요일(공휴일)인 케이스는 클러스터 2로 배정되었음을 확인할 수 있다. 이는 평일 그룹의 클러스터 2 설명인 "평일 공휴일 낮 시간대에 승차가 많은 곳 리스트"와 일치하며, 공휴일임을 반영하여 적절한 추천이 이루어졌다고 해석할 수 있다. 또한, 2023년 11월 29일 수요일 저녁 20시의 케이스에서는 클러스터 1로 배정되었는데, 이는 "평일 비 내리는 저녁 시간대에 승차가 많은 곳 리스트"와 일치하여 해당 시간대에 적절한 추천이 이루어졌음을 확인할 수 있다.

	날짜	요일	위도	경도	클러스터	정답률 퍼센트
0	2023-05-29 20:00:00	Monday	36.351823	127.385284	2	50.0
1	2023-05-29 20:00:00	Monday	36.327489	127.405289	2	50.0
2	2023-05-29 20:00:00	Monday	36.333528	127.409182	2	50.0
3	2023-05-29 20:00:00	Monday	36.347092	127.386542	2	50.0
4	2023-05-29 20:00:00	Monday	36.359872	127.387291	2	50.0

<표 15> 결과 해석 – 2023년 5월 29일 월요일 20시

241	2023-11-29 20:00:00	Wednesday	36.352761	127.404273	1	50.0
242	2023-11-29 20:00:00	Wednesday	36.355481	127.380293	1	50.0
243	2023-11-29 20:00:00	Wednesday	36.328694	127.390572	1	50.0
244	2023-11-29 20:00:00	Wednesday	36.334521	127.401983	1	50.0

<표 16> 결과 해석 – 2023년 11월 29일 수요일 20시

3. 결 론

1. 기대효과

본 프로젝트에서 개발한 이동 위치 추천 시스템은 택시 기사들이 수익을 극대화할 수 있도록 효율적인 운행 경로를 제안한다.

택시 수요가 높은 지역을 실시간으로 추천하여 기사들이 보다 효율적으로 운행할 수 있게 해 택시 기사 수익을 증대한다. 불필요한 공차 시간을 줄이고, 높은 수익을 창출할 수 있다. 승객이 많은 장소와 시간대에 택시가 보다 빠르게 도착함으로써 승객들의 대기 시간이 단축된다. 이는 승객 대기 시간을 감소해 승객의 서비스 만족도를 높이는 데 기여할 것이다.

요일, 시간, 날씨 등 다양한 변수에 맞춘 최적화된 이동 경로를 제시함으로써 전체적인 택시 운송 효율이 향상된다. 이를 통해 대전시의 교통 흐름을 개선하고, 택시 자원의 효율적 운영을 도울 수 있다.

기상 정보, 공휴일, 관광지 정보 등 다양한 데이터를 결합하여 보다 정교한 분석과 예측이 가능해졌으며, 향후 더욱 다양한 요인들을 통합해 활용할 수 있는 확장 가능성이 있다.

2. 개선 방향

현재 시스템은 택시 기사들의 운행 데이터를 기반으로 클러스터링을 수행하여 최적의 이동 경로를 추천하고 있다. 기사 성향 클러스터링 결과를 추천 시스템의 가중치에 반영함으로써, 각 기사별 맞춤형 추천이 가능하도록 발전시킬 수 있는 가능성을 확인했다. 예를 들어, 특정 기사가 장거리 운행을 선호하거나, 할증 운행이 많은 지역을 자주 운행하는 패턴이 확인될 경우, 해당 성향에 맞는 경로와 승객 위치를 우선 추천하는 방식으로 서비스를 개선할 수 있다. 현재는 차량 이름을 기준으로 데이터를 그룹화하고, 평균 승차 거리, 평균 요금, 평균 할증 여부, 평균 주행 시간을 계산하여 기사들의 운행 성향으로 클러스터링하고 변수 중요도도로 표현하는 것에 그쳤지만, 이를 추천 시스템에 적용한다면 지금보다 더 개선된 정확도와 개인화된 서비스를 제공할 수 있을 것이라 생각한다.

또한 열차시간표를 활용하지 못하였는데, 열차역 주변의 승차 수요는 열차 도착 및 출발 시간에 크게 좌우되므로, 만약 시간표 데이터를 반영한다면, 열차역 주변에서의 승객 수요 패턴을 예측했을때때 더욱 정확한 추천이 가능할 것이다. 이를 통해 열차역 근처에서의 높은 승차 수요 시간을 효율적으로 포착하고, 기사들에게 보다 유용한 정보를 제공할 수 있을 것으로 기대된다.

현재는 위치, 시간대, 요일, 날씨, 공휴일 여부를 입력했을 때 클러스터링 결과에 따라 10개의 추천 좌표를 제공하고 지도에 표시하는 것까지만 진행하였지만, 이동 추천 위치를 음성으로 안내해주는 앱이나 모바일 웹 서비스를 개발하여 더 직관적이고 편리한 사용자 경험을 제공할 수 있을 것이다. 추후 이러한 앱 개발을 통해 사용자들이 시스템을 보다 쉽게 이용할 수 있도록 하는 것이 앞으로의 개선 방향이다.

3. 활용 데이터 출처

1) 한국 공휴일 API : 공공 데이터 포털

1) 종관기상관측(ASOS): 기상청 기상자료개방 포털
2) 대전광역시 택시 승강장 현황: 공공 데이터 포털
3) 대전광역시 인기 관광지 현황: 한국관광 데이터랩