부산대학교
PUSAN NATIONAL UNIVERSITY

[데이터마이닝 과제]

■과 목 명	데이터마이닝
■담당교수	권준호 교수님
■제 출 일	2024.12.28
■학　과	정보컴퓨터공학부
■학　번	202155650
■성　명	윤소현

1. 데이터셋 준비

데이터 분석 과제를 진행하기 위해 가장 먼저 국내 공공 데이터 포털 https://www.data.go.kr/에서 분석할 데이터셋을 찾아 보았다. 분석뿐만 아니라 데이터마이닝 강의 시간에 배운 logistic regression도 적용해서 모델을 만들어보고 예측 정확도도 확인해 볼 수 있을만한 데이터셋을 선택하고자 했다. 따라서 이번 과제를 위해 한국 사회 보장 정보원에서 제공하는 전국 시군구별 어린이집 연장 보육반 운영 여부 데이터셋을 선택했다. 272,665 개의 행으로 이루어져 있어 간단한 regression 모델을 만들어 보기에도 적절하다고 생각되었다.

이 데이터는 2017년부터 2024년까지 전국 시군구별 어린이집 별 운영 중인 어린이집 공공 데이터로, 연도, 시도, 시군구, 어린이집 유형, 어린이집 명, 정원, 현원, 연장보육반 운영 여부를 확인할 수 있다. 연장 보육은 16시부터 19시 30분까지 제공되는 보육 서비스로 2020년부터 시행하였다고 한다. 이 점을 참고해서 분석을 진행해 보았다.

2. 데이터 분석 목적

전국 시군구별 어린이집의 연장보육반 운영 여부를 분석하고, 로지스틱 회귀 모델을 활용하여 어린이집이 연장보육반을 운영할 가능성을 예측해 본다.

3. 데이터 분석 및 시각화를 위한 라이브러리 불러오기

```
import pandas as pd #1
import matplotlib.pyplot as plt #2
import seaborn as sns #3
```

#1) 가장 먼저 csv 파일을 읽어올 수 있게끔 pandas 라이브러리를 불러왔다. pandas 라이브러리는 추후에 데이터 전처리를 위해서도 사용된다.

#2) 데이터 시각화를 위해 matplotlib의 pyplot 모듈을 불러왔다.

#3) seaborn 라이브러리는 matplotlib를 기반으로 하는 고급 시각화 라이브러리이다. 이를 이용해서 데이터를 보다 직관적으로 시각화할 수 있게 된다.

4. 데이터 전처리

```
#1
kindergarden = pd.read_csv("/content/drive/MyDrive/한국사회보장정보원_전국 시군구별 어린이집 연장보육반 운영 여부_20241130.csv", encoding='cp949')

#2
kindergarden['연장보육반운영여부'] = kindergarden['연장보육반운영여부'].map({'Y': 1, 'N': 0})

#3
kindergarden['어린이집유형'] = kindergarden['어린이집유형'].map({
    '가정': 'Home',    '국공립': 'Public',    '민간': 'Private',
    '사회복지법인': 'Social Welfare Corporation',    '직장': 'In the company',
})

#4
```

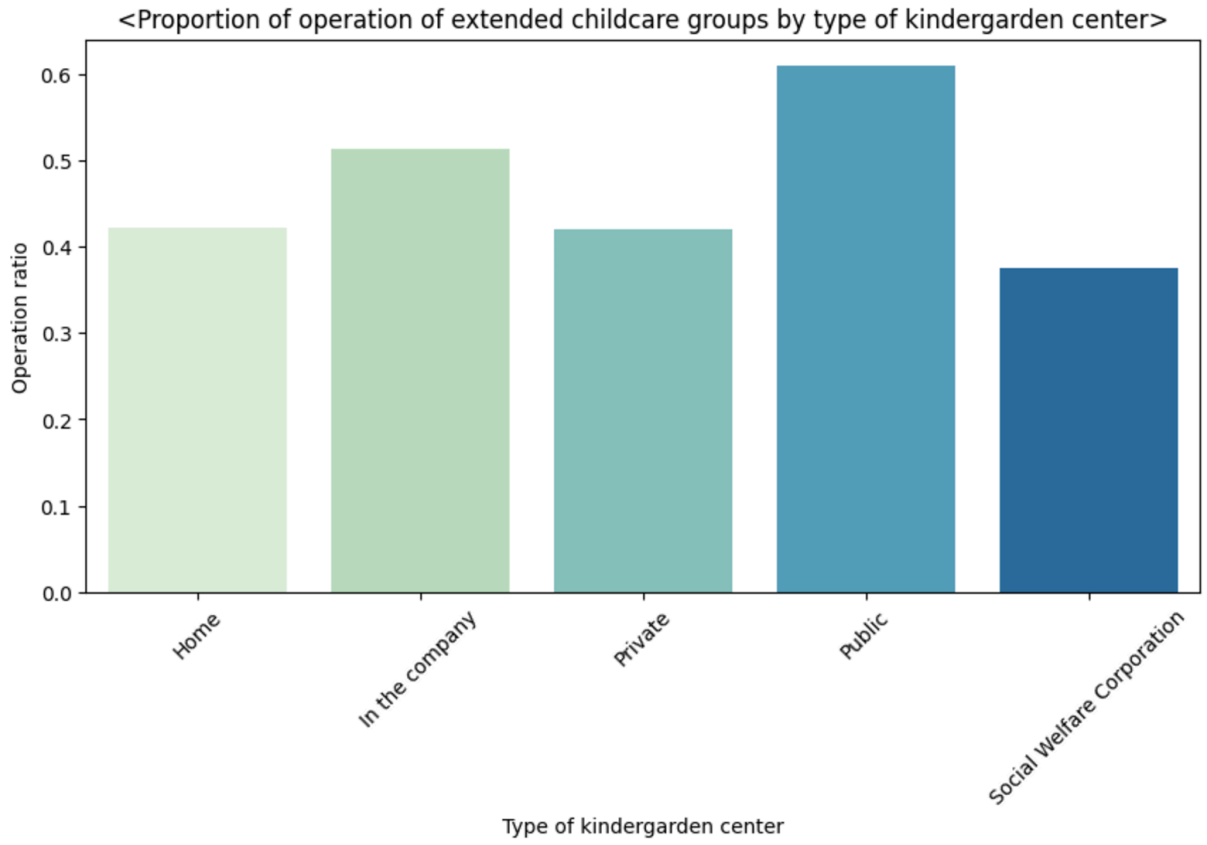
```
# '시도' 컬럼 값 영어로 변경
kindergarden['시도'] = kindergarden['시도'].map({
    '서울특별시': 'Seoul',    '부산광역시': 'Busan',    '대구광역시': 'Daegu',
    '인천광역시': 'Incheon',    '광주광역시': 'Gwangju',    '대전광역시': 'Daejeon',
    '울산광역시': 'Ulsan',    '세종특별자치시': 'Sejong',    '경기도': 'Gyeonggi',
    '강원특별자치도': 'Gangwon',    '충청북도': 'Chungbuk',    '충청남도': 'Chungnam',
    '전북특별자치도': 'Jeonbuk',    '전라남도': 'Jeonnam',    '경상북도': 'Gyeongbuk',
    '경상남도': 'Gyeongnam',    '제주특별자치도': 'Jeju'
})
```

#1) 먼저 pandas의 read_csv() 함수로 csv 파일을 읽어들여 DataFrame 객체로 변경해 주었다.

#2) csv 파일 내에는 연장 여부를 'Y' / 'N'으로 표시하고 있으므로 데이터 처리를 용이하게 하기 위해 map 함수를 사용하여 'Y'를 1로, 'N'을 0으로 변환해 주었다.

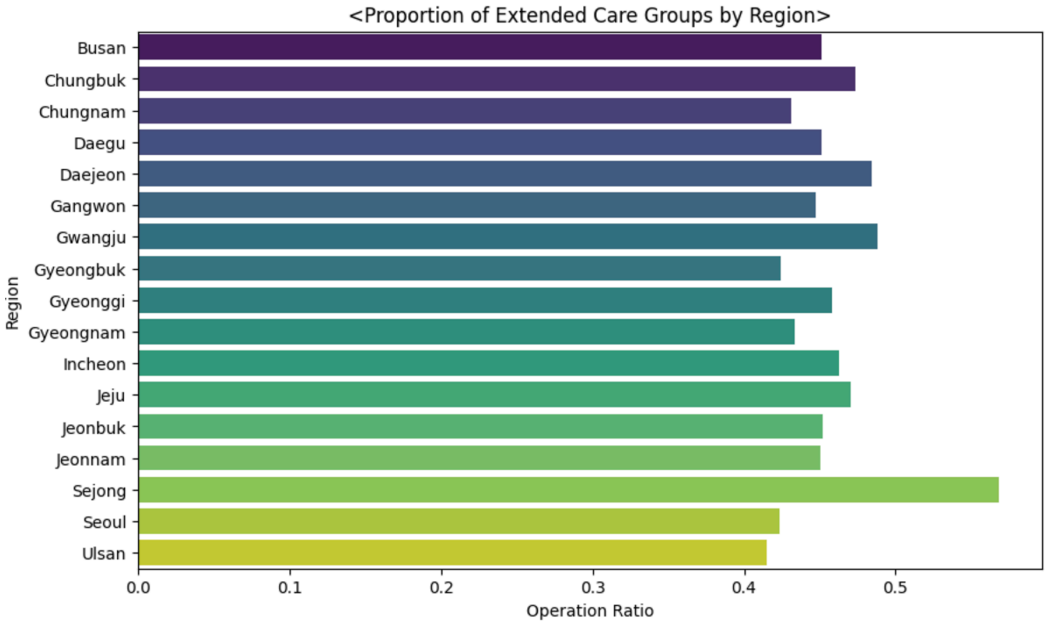
#3, #4) csv 파일 내의 value 값들이 한글로 되어있는 경우, 데이터 시각화 시에 한글 깨짐 문제가 발생하여 영어 단어로 매핑하는 작업을 거쳤다.

5. 어린이집 유형별 연장 보육 운영 비율 분석



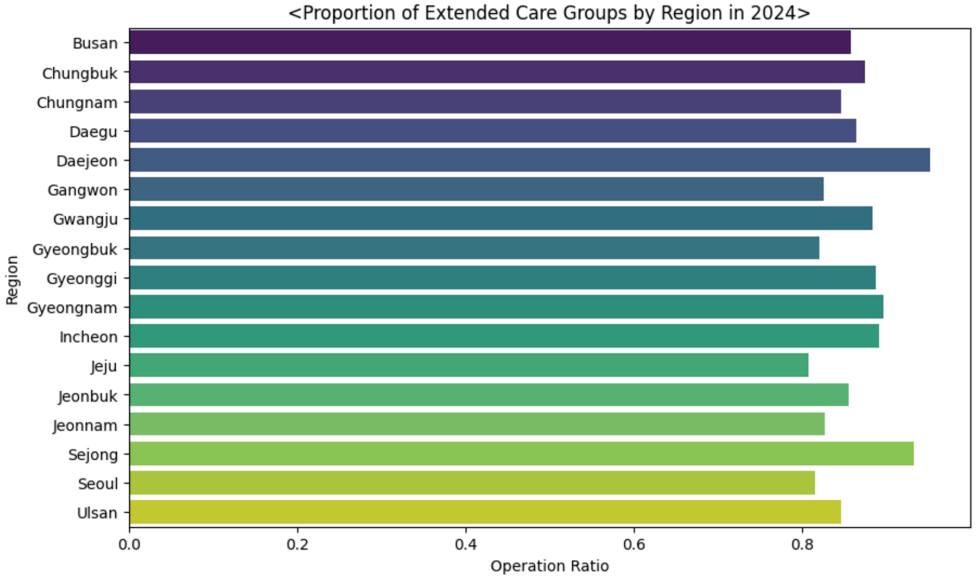
어린이집 유형별로 연장 보육을 운영하고 있는 비율을 시각화한 결과이다. Public (국공립 어린이집) 유형의 어린이집이 가장 높은 비율로 연장 보육을 운영하고 있는 것을 확인할 수 있다. 반대로 Social Welfare Corporation (사회복지법인) 유형의 어린이집이 가장 작은 비율로 연장 보육을 운영하고 있다.

6. 시도별 연장 보육 운영 비율 분석



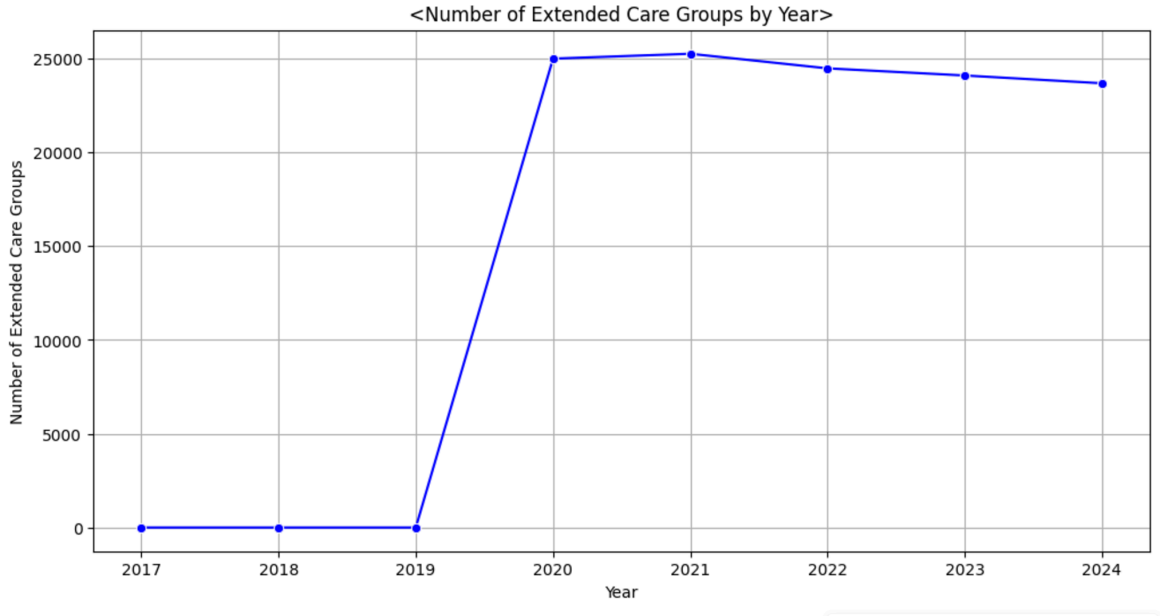
전체 기간 동안의 시/도별 연장 보육 운영 비율을 시각화한 그래프를 보면, 세종시가 가장 높은 비율로 연장 보육을 운영했던 것을 알 수 있다.

7. 2024년 시도별 연장 보육 운영 비율 분석



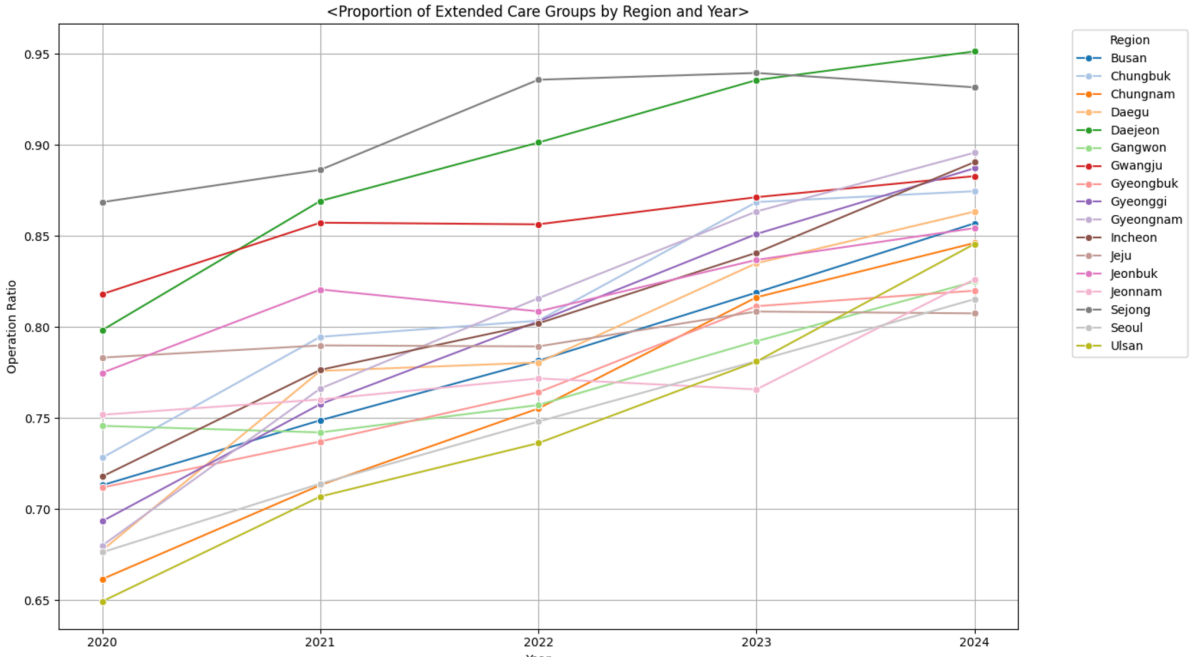
2024년 올 한해를 기준으로 시/도별 연장 보육을 운영 비율을 살펴보면, 대전광역시와 세종특별자치시가 가장 높은 비율로 연장 보율을 운영하고 있고, 제주특별자치도에서 가장 낮은 비율로 운영되고 있다. 분석에 포함된 모든 시/도의 연장 보육 비율이 60%를 훌쩍 넘는 것도 확인할 수 있다.

8. 연도별 연장 보육 운영 어린이집 개수 분석



어린이집 연장 보육이 시작된 2020년, 그리고 그 다음 해 최고 개수를 달성하고, 점점 줄어드는 추세인 것을 확인할 수 있었다.

9. 연도별 각 시도의 연장 보육 운영 비율



연도별 연장 보육 운영 비율 추이를 분석해 보면, 2020년도에는 세종특별자치시, 광주광역시, 대전광역시 순으로 가장 높은 비율로 연장 보육을 운영하였으나 대전광역시의 꾸준한 운영 비율 상승으로 2024년에는 가장 높은 비율로 연장 보육을 운영하는 지역이 되었음을 확인할 수 있었다.

10. Logistic regression 모델 만들어서 예측 성능 측정해 보기

데이터셋의 연장 보육 운영 여부가 0 또는1로 표시되기 때문에 이번 학기 데이터마이닝 강의 시간에 배운 logistic regression을 이용해 간단한 분류 모델을 생성하고, 예측 결과를 측정해 보고자 한다.

```
from sklearn.model_selection import train_test_split #1
from sklearn.preprocessing import StandardScaler #2
from sklearn.linear_model import LogisticRegression #3
from sklearn.metrics import accuracy_score #4
```

파이썬에서 사용할 수 있는 관련 라이브러리들을 먼저 불러왔다.

#1) 전체 데이터를 training data / validation data / test data로 분리하여 학습을 진행하기 위해서 train_test_split 함수를 import 했다.

#2) feature data의 스케일을 표준화하여 모델 성능을 개선하기 위해 해당 클래스를 import 했다.

#3) 로지스틱 회귀 모델 클래스를 import 하고,

#4) 모델의 정확도를 계산하기 위한 함수를 불러왔다.

```
kindergarden = pd.get_dummies(kindergarden, columns=['시도', '어린이집유형'],
drop_first=True)

# 특성 변수와 타겟 변수 분리
X = kindergarden[['연도', '정원', '현원'] + [col for col in kindergarden.columns if
col.startswith('시도_') or col.startswith('어린이집유형_')]]
y = kindergarden['연장보육반운영여부']

# 전체 데이터의 60%를 training data, 20%를 validation data, 20%를 test data로 설정
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4,
random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)

# 표준화
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# 모델 훈련 - 로지스틱 회귀
model = LogisticRegression(random_state=42)
model.fit(X_train, y_train)

# 예측
y_pred = model.predict(X_test)

# 성능 평가
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

간단히 생성해 본 모델의 정확도로는 83%가 출력되었다.

```
Accuracy: 0.83
```

11. 구글 코랩 링크

https://colab.research.google.com/drive/1eIoOpWrSR_FrdXwg1Uhk405253IzQv2-?usp=sharing