Multivariate Data Analysis Assignment #5

Association Rule & Clustering

산업경영공학부 2020170856 이우진

[Part 1: Association Rule Mining]

Dataset : **MOOC Dataset (big_student_clear_third_version.csv)**

해당 데이터셋은 MOOC 강좌를 수강한 수강생들에 대한 정보가 포함되어 있는 데이터 셋이다. 다음 각 Instruction에 따라 데이터를 변환하고 연관규칙분석을 수행하여 각 결과물을 제시하고 적절한 해석을 제공하시오.

[Step 1] 데이터 변환

[Q1] 원 데이터는 총 416,921건의 관측치와 22개의 변수가 존재하는 데이터프레임이다. 이 중에서 아래 그림과 같이 userid_DI (사용자 아이디)를 Transaction ID로 하고, institute (강좌 제공 기관), course_id (강좌코드), final_cc_cname_DI (접속 국가), LoE_DI (학위 과정)을 하나의 string으로 결합하여 Item Name으로 사용하는 연관규칙 분석용 데이터셋을 만드시오.

원 데이터는 총 416,921개의 row와 22개의 columns을 가진 데이터로 먼저, 여러 개의 변수를 하나의 string으로 결합하기 전, 결측치가 있는지 확인해보았다.

```
# 결측치 확인
missing_values = df.isnull().sum()

# 결측치가 있는 열 출력
print("결측치가 있는 열:")
print(missing_values[missing_values > 0])
```

```
결측치가 있는 열:
gender    23211
dtype: int64
```

다음과 같이 gender 변수에 결측치들이 존재하는 것을 알 수 있는데, 해당 변수는 Transaction ID, Item Name으로 사용되는 데이터가 아니므로 무시하였다.

변수 중, userid_DI를 Transaction ID로 하고, institute, course_id, final_cc_cname_DI, LoE_DI를 하나의 string으로 결합하여 Item Name으로 사용하는 연관규칙 분석용 데이터셋을 만들었다.

이때, Item Name은 각각의 변수 값을 '_' 기호를 사용하여 연결하였고, groupby 함수를 통해 userid_DI별로 그룹화하여 Transaction ID와 Item Name를 만들었다.

	userid_DI	Item Name
0	MHxPC130000002	MITx_14.73x_United Kingdom_Secondary
1	MHxPC130000004	HarvardX_CS50x_India_Secondary,HarvardX_ER22x_...
2	MHxPC130000006	HarvardX_ER22x_United States_Bachelor's
3	MHxPC130000007	HarvardX_CB22x_United States_Master's
4	MHxPC130000008	MITx_6.00x_United Kingdom_Bachelor's
...	...	...
335645	MHxPC130597670	HarvardX_CS50x_United States_Secondary
335646	MHxPC130597671	MITx_6.00x_India_Master's
335647	MHxPC130597672	MITx_6.00x_Bangladesh_Secondary
335648	MHxPC130597674	HarvardX_PH207x_Germany_Doctorate
335649	MHxPC130597675	MITx_6.00x_Other Africa_Bachelor's

[335650 rows x 2 columns]

다음과 같이 총 335,650개의 rows와 2개의 columns이 만들어진 것을 확인할 수 있다. 행의 개수가 줄어든 이유는 groupby 함수로 인해, userid_DI 하나에 Item Name이 여러 개가 들어갈 수 있기 때문에 전체적인 행의 개수가 줄어든 것이다.

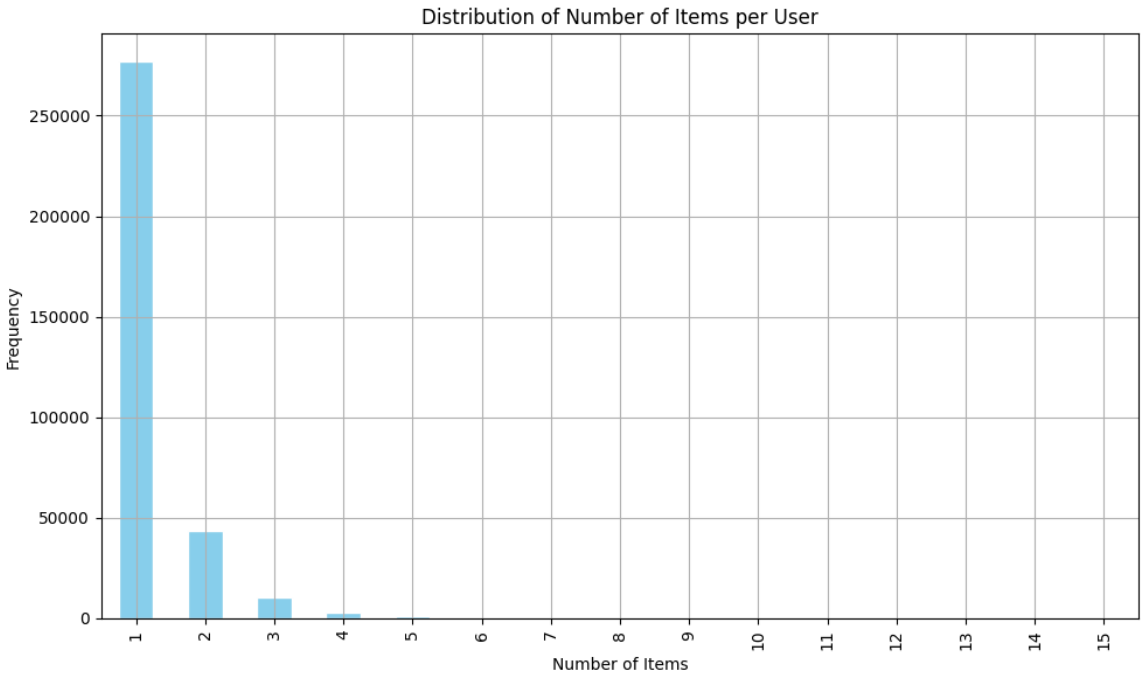
결과적으로 Transaction ID와 Item Name가 잘 만들어진 것을 확인할 수 있다.

[Step 2] 데이터 불러오기 및 기초 통계량 확인

[Q2-1] [Q1]에서 생성된 데이터를 읽어들이고 해당 데이터에 대한 탐색적 데이터 분석을 수행하여 데이터의 특징을 파악해보시오.

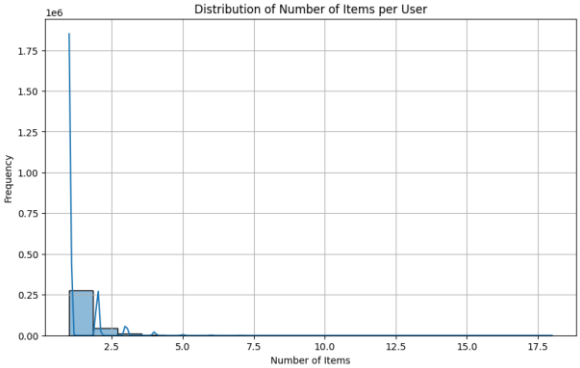
생성된 데이터에 결측치가 있는지 확인한 결과, 결측치가 없는 것을 알 수 있었다.

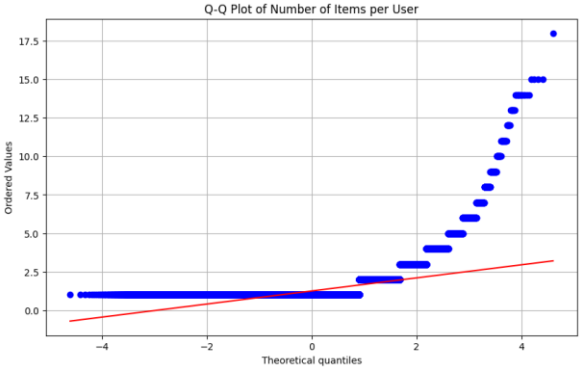
다음으로 각 사용자별로 수강한 아이템 수를 계산하여, 막대 그래프로 시각화 하였고, 그 결과는 아래와 같다.



위의 그래프를 보았을 때, 대다수의 사용자가 1개의 강의를 수강하고, 2~4개의 강의를 듣는 사용자도 눈에 띄게 있는 것을 알 수 있다. 최대 15개의 강의를 듣는 사람이 있는 것도 확인하였다.

각 사용자별로 수강한 아이템 수가 정규성을 따르는지, Shapiro-Wilk 검정을 수행한 결과, p-value가 0.000으로 귀무가설이 기각되어, 정규 분포를 따르지 않는다는 것을 알 수 있다. 이와 관련하여 아래 히스토그램과 Q-Q 플롯도 확인해보았는데, 정규성을 띄지 않는 것을 알 수 있다.

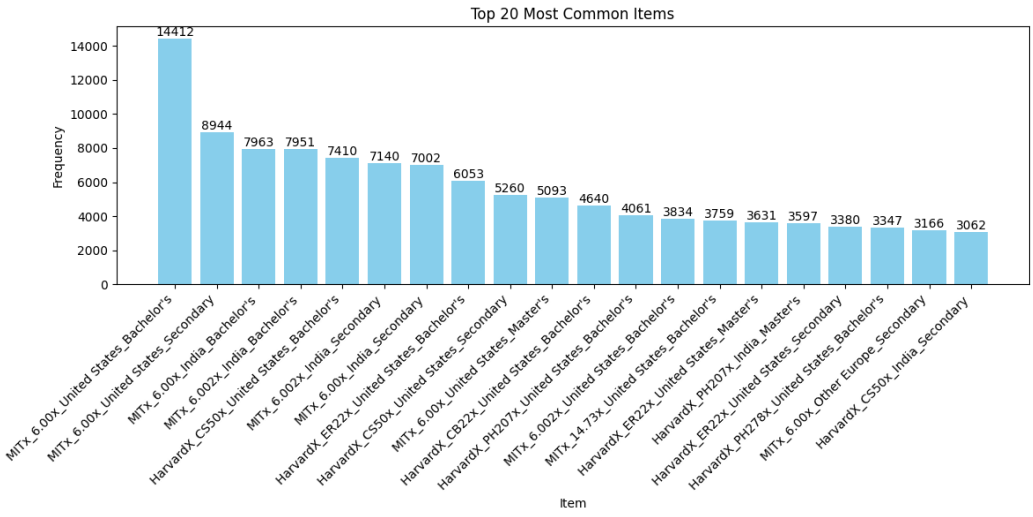




describe()를 통해, 통계를 확인해보았고, 그 결과는 다음과 같다. 즉, 평균적으로 사용자 1명당 약 1.24개의 강의를 수강하고, 그 표준편차는 약 0.63임을 알 수 있다.

```
count    335650.000000
mean          1.242130
std           0.630132
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
max          15.000000
Name: Item Name, dtype: float64
```

다음으로 각 아이템의 빈도를 계산하여, 막대그래프로 시각화 하려고 하였는데, 전체 아이템의 개수를 확인해보니, 총 1405개의 강좌가 있는 것을 확인할 수 있었다. 따라서 이 중, 가장 높은 빈도를 보이는 상위 20개 아이템을 추출하여 막대그래프로 나타내 보았고, 그 결과는 아래와 같다.



MITx_6.00x_United States_Bachelor's의 빈도 수가 14412로 가장 높은 것을 확인할 수 있고, 다른 강좌들의 빈도 수보다 눈에 띄게 높은 것을 확인할 수 있다. 그 다음으로, MITx_6.00x_United States_Secondary, MITx_6.00x_India_Bachelor's, MITx_6.002x_India_Bachelor's 순으로 빈도 수가 높은 것을 알 수 있다.

[Q2-2] 아이템 이름과 아이템 카운트를 이용하여 워드클라우드를 생성해 보시오.

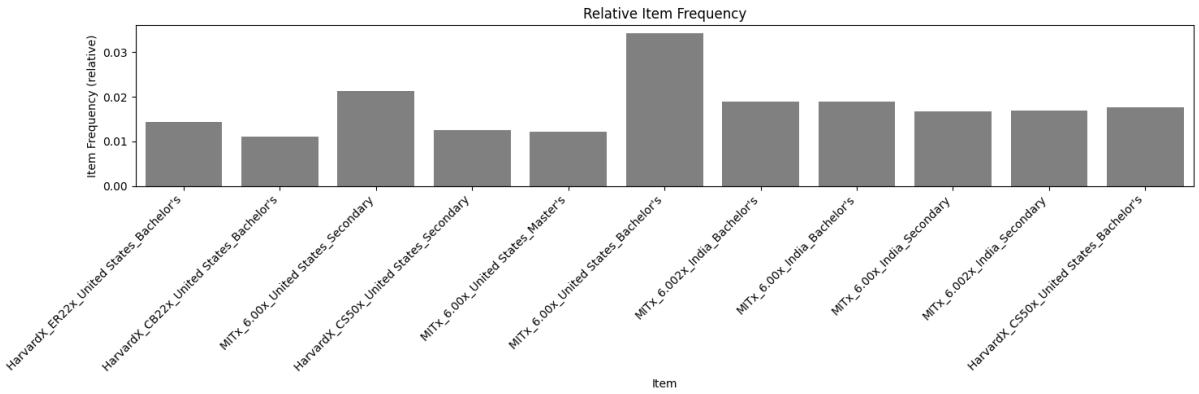
아이템 이름과 카운트를 활용하여 워드클라우드를 생성한 결과는 다음과 같다.

워드클라우드는 텍스트 데이터를 시각화하여 가장 빈번하게 등장하는 단어들을 강조하는 도구이다. 이때, 워드클라우드에서 큰 글씨로 표시된 단어는 데이터에서 자주 등장한 단어를 나타낸 것

으로, 가장 큰 글씨로 표시된 단어가 데이터셋에서 가장 빈번하게 등장한 단어이다. 결과적으로 MITx_6.00x_United States_Bachelor's의 빈도 수가 가장 높은 것을 워드클라우드에서도 확인이 가능하다. 또한, 빈도 수가 높았던 MITx_6.00x_United States_Secondary, MITx_6.00x_India_Bachelor's, MITx_6.002x_India_Bachelor's도 다른 것들에 비해 큰 글씨로 표현되어 있는 것을 확인할 수 있다.

[Q2-3] 최소 빈도 1% 이상 등장한 Items들의 Bar Chart를 도시하시오. 상위 5개의 Item에 대해 접속 국가는 각각 어느 국가인지 확인하시오.

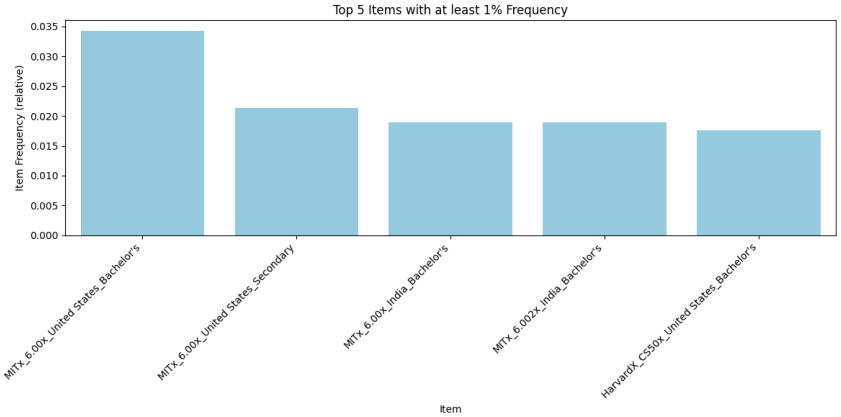
최소 빈도 1% 이상 등장한 Items들의 Bar Chart를 도시한 결과는 아래와 같다.



총 11개의 Item이 빈도가 1% 이상인 것을 확인할 수 있다. 이때, 각 아이템의 빈도 수와 비율은 아래와 같다.

item	freq	ratio
HarvardX_ER22x_United States_Bachelor's	6053	0.014518
HarvardX_CB22x_United States_Bachelor's	4640	0.011129
MITx_6.00x_United States_Secondary	8944	0.021453
HarvardX_CS50x_United States_Secondary	5260	0.012616
MITx_6.00x_United States_Master's	5093	0.012216
MITx_6.00x_United States_Bachelor's	14412	0.034568
MITx_6.002x_India_Bachelor's	7951	0.019071
MITx_6.00x_India_Bachelor's	7963	0.019100
MITx_6.00x_India_Secondary	7002	0.016795
MITx_6.002x_India_Secondary	7140	0.017126
HarvardX_CS50x_United States_Bachelor's	7410	0.017773

이 중, 상위 5개의 Item에 대해 Bar Chart를 도시한 결과는 아래와 같다.



이때, 각 아이템의 빈도 수와 비율은 아래와 같다.

	item	freq	ratio
35	MITx_6.00x_United States_Bachelor's	14412	0.034568
16	MITx_6.00x_United States_Secondary	8944	0.021453
37	MITx_6.00x_India_Bachelor's	7963	0.019100
36	MITx_6.002x_India_Bachelor's	7951	0.019071
72	HarvardX_CS50x_United States_Bachelor's	7410	0.017773

상위 5개의 Item에 대한 접속 국가는 각각 'United States', 'United States', 'India', 'India', 'United States'인 것을 알 수 있다. 상위 5개의 item이 아닌 최소 빈도 1%를 만족하는 11개의 item의 접속 국가를 확인하여도, 'United States' 또는 'India'인 것을 알 수 있다. 이는 미국과 인도에서 해당 강좌에 대한 수요가 높다는 것을 의미한다.

[Step 3] 규칙 생성 및 결과 해석

[Q3-1] 최소 10개 이상의 규칙이 생성될 수 있도록 support와 confidence의 값을 조정해 가면서 각 support-confidence 조합에 대해 총 몇 가지의 규칙이 생성되는지 확인하고 그 결과를 아래 표와 같은 형태로 제시하시오. 최소한 3개 이상의 support, 3개 이상의 confidence, 총 9개 이상의 조합에 대한 규칙 생성을 수행하시오.

10개 이상의 규칙이 생성될 수 있도록 아래와 같이 support, confidence의 조합을 만들었다. 처음에 support values를 [0.003, 0.002, 0.001]으로 설정하였지만, support_values가 0.003일 때, 최소 규칙 10개 이상이라는 조건을 만족하지 못했기 때문에 제거하고 아래와 같이 수정하였다.

min_support_values = [0.002, 0.0015, 0.001] # Minimum support thresholds

min_confidence_values = [0.05, 0.03, 0.01, 0.005, 0.001] # Minimum confidence thresholds

위의 조합으로 규칙 생성을 수행해보았고, 그 결과는 아래와 같다.

Number of rules	Confidence = 0.05	Confidence = 0.03	Confidence = 0.01	Confidence = 0.005	Confidence = 0.001
Support = 0.0020	20	20	20	20	20
Support = 0.0015	29	30	30	30	30
Support = 0.0010	51	55	56	56	56

Support는 규칙에 포함된 아이템이 전체 거래 중 얼마나 자주 나타나는지를 나타내는 지표이다.

Support 임계 값을 낮출수록 더 많은 규칙이 발견되는 것을 알 수 있다. 이는 덜 빈번한 아이템 셋도 규칙으로 포함되기 때문에 나타나는 결과이다.

Confidence는 연관 규칙의 조건이 참일 때 결과도 참일 확률을 나타낸다. Confidence 임계 값을 낮출수록 더 많은 규칙이 발견되는 것을 알 수 있다. 이는 덜 신뢰할 수 있는 규칙도 포함되기 때문에 나타나는 결과이다.

전체적으로 Support와 Confidence 임계 값을 낮출수록 발견되는 규칙의 수가 증가하는 것을 알 수 있다. 그러나, Support 0.002에서는 모든 Confidence 값에서 20개의 규칙이 발견되는 것을 알 수 있고, Support 0.0015에서도 confidence가 0.05에서 0.03으로 감소할 때만 규칙의 수가 증가하고, 그 이후로는 규칙의 수에 변화가 없는 것을 알 수 있다.

이는, Support 0.002일 때, 빈번한 항목 집합에서 가능한 모든 규칙이 20개이며, 따라서 confidence 값이 달라지더라도 추가적인 규칙이 생성되지 않는 것이라고 해석이 가능하다. 다른 경우들도 마찬가지로, 어느 임계점 이후에는 가능한 모든 규칙이 이미 생성되어 신뢰도 임계 값이 변화하더라도 추가적인 규칙이 나오지 않는다고 할 수 있다.

[Q3-2] support = 0.001, confidence = 0.05로 지정하여 생성된 연관규칙분석들에 대해 다음 질문에 대한 답과 본인의 생각을 서술하시오.

support = 0.001, confidence = 0.05로 지정하여 연관규칙분석들을 생성해 보았고, 총 678 combinations으로 51개의 규칙들이 만들어진 것을 확인할 수 있다.

1. Support가 가장 높은 규칙은 무엇인가?

Antecedents가 (MITx_6.00x_United States_Bachelor's)이고, consequents가 (HarvardX_CS50x_United States_Bachelor's)인 규칙이 support가 0.003644로 가장 높다. 결과를 표로 나타낸 것은 아래와 같다.

antecedent support	consequent support	support	confidence	lift	leverage	conviction	Zhangs metric
0.042282	0. 022077	0.003644	0.086175	3.903474	0.002710	1.070143	0.776657

Support가 0.003644인 것은, 전체 데이터에서 두 강의를 모두 수강한 사용자의 비율이 약 0.364%인 것을 의미한다.

Confidence를 보았을 때, MITx_6.00x_United States_Bachelor's 강의를 수강한 사용자가 HarvardX_CS50x_United States_Bachelor's 강의를 수강할 확률이 약 8.6%인 것을 의미한다.

Lift를 보았을 때, 'HarvardX_CS50x_United States_Bachelor's' 과정을 수강한 사용자가

'MITx_6.00x_United States_Bachelor's' 과정을 수강할 확률이 일반적인 경우보다 약 3.9배 높다는 것을 것이고, 이는 두 강의 간의 상관 관계가 매우 높음을 의미한다.

Leverage는 두 과정 간의 규칙이 무작위로 발생할 경우와 비교하여 얼마나 더 자주 발생하는지 나타내는 값으로, 양의 값은 상관 관계가 있음을 의미한다. 0.002710 값은 두 과정 간의 양의 상관 관계가 있음을 나타낸다.

Conviction은 'HarvardX_CS50x_United States_Bachelor's' 과정을 수강한 사용자가 'MITx_6.00x_United States_Bachelor's' 과정을 수강하지 않을 확률이 약 1.07배 줄어듦을 의미한다.

Zhang's Metric은 1에 가까울수록 강한 상관 관계를 의미하고, 따라서 0.776657 값은 두 과정 간의 강한 상관 관계를 나타낸다.

Antecedents가 (HarvardX_CS50x_United States_Bachelor's)이고, consequents가 (MITx_6.00x_United States_Bachelor's)인 규칙도 위와 동일한 support 값을 가진다. 즉, 조건절과 결과절의 순서가 바뀌더라도, support 값은 같기 때문에 이러한 결과가 나왔다고 할 수 있다. 이 경우, 나머지 수치들에서는 큰 차이가 없지만 confidence 값은 0.165047로, HarvardX_CS50x_United States_Bachelor's 강의를 수강한 사용자는 약 16.5% 확률로 MITx_6.00x_United States_Bachelor's 강의를 수강한다고 할 수 있다. 이는 조건절과 결과절이 반대인 경우의 2배이기 때문에 큰 차이라고 할 수 있고, support와 lift는 동일하기 때문에, 해당 규칙이 앞의 규칙보다 더 좋은, 유용한 규칙이라고 할 수 있다.

2. Confidence가 가장 높은 규칙은 무엇인가?

Antecedents가 (MITx_8.02x_India_Secondary)이고, consequents가 (MITx_6.002x_India_Secondary)인 규칙이 confidence가 0.388109로 가장 높다. 결과를 표로 나타낸 것은 아래와 같다.

antecedent support	consequent support	support	confidence	lift	leverage	conviction	Zhangs metric
0.007216	0.020414	0.002801	0.388109	19.011790	0.002653	1.600916	0.954287

Support가 0. 002801인 것은, 전체 데이터에서 두 강의를 모두 수강한 사용자의 비율이 약 0.28%인 것을 의미한다.

Confidence를 보았을 때, MITx_8.02x_India_Secondary 강의를 수강한 사용자가 MITx_6.002x_India_Secondary 강의를 수강할 확률이 약 38.8%인 것을 의미한다. 이는 MITx_8.02x_India_Secondary 강의를 수강한 1/3 이상의 학생들이 MITx_6.002x_India_Secondary를 듣는다는 것으로, 유용한 규칙이라고 할 수 있다.

Lift를 보았을 때, MITx_8.02x_India_Secondary 과정을 수강한 사용자가

MITx_6.002x_India_Secondary 과정을 수강할 확률이 일반적인 경우보다 약 19배 높다는 것을 것이고, 이는 두 강의 간의 상관 관계가 매우 높음을 의미한다.

Leverage는 두 과정 간의 규칙이 무작위로 발생할 경우와 비교하여 얼마나 더 자주 발생하는지 나타내는 값으로, 양의 값은 상관 관계가 있음을 의미한다. 0. 002653값은 두 과정 간의 양의 상관 관계가 있음을 나타낸다.

Conviction은 MITx_8.02x_India_Secondary 과정을 수강한 사용자가 MITx_6.002x_India_Secondary 과정을 수강하지 않을 확률이 약 1.6배 줄어듦을 의미한다.

Zhang's Metric은 1에 가까울수록 강한 상관 관계를 의미하고, 따라서 0. 954287 값은 두 과정 간의 강한 상관 관계를 나타낸다.

결론적으로, 두 강의 모두 India 국가이고, MIT 강의이며, 학위 과정이 동일한 Secondary라는 점에서 두 강의를 같이 들을 확률이 높고, 따라서 이러한 결과가 나왔다고 해석할 수 있다.

3. Lift가 가장 높은 규칙은 무엇인가?

Antecedents가 (MITx_8.02x_United States_Bachelor's)이고, consequents가 (MITx_6.002x_United States_Bachelor's)인 규칙이 Lift가 0.388109로 가장 높다. 결과를 표로 나타낸 것은 아래와 같다.

antecedent support	consequent support	support	confidence	lift	leverage	conviction	Zhangs metric
0.006435	0.011059	0.001391	0.216204	19.549777	0.001320	1.261732	0.954994

Support가 0. 001391인 것은, 전체 데이터에서 두 강의를 모두 수강한 사용자의 비율이 약 0.14%인 것을 의미한다.

Confidence를 보았을 때, MITx_8.02x_United States_Bachelor's 강의를 수강한 사용자가 MITx_6.002x_United States_Bachelor's 강의를 수강할 확률이 약 21.6%인 것을 의미한다.

Lift를 보았을 때, MITx_8.02x_United States_Bachelor's 과정을 수강한 사용자가 MITx_6.002x_United States_Bachelor's 과정을 수강할 확률이 일반적인 경우보다 약 19.5배 높다는 것을 것이고, 이는 두 강의 간의 상관 관계가 매우 높음을 의미한다.

Leverage는 두 과정 간의 규칙이 무작위로 발생할 경우와 비교하여 얼마나 더 자주 발생하는지 나타내는 값으로, 양의 값은 상관 관계가 있음을 의미한다. 0. 001320값은 두 과정 간의 양의 상관 관계가 있음을 나타낸다.

Conviction은 MITx_8.02x_United States_Bachelor's 과정을 수강한 사용자가 MITx_6.002x_United States_Bachelor's 과정을 수강하지 않을 확률이 약 1.26배 줄어듦을 의미한다.

Zhang's Metric은 1에 가까울수록 강한 상관 관계를 의미하고, 따라서 0. 954994값은 두 과정 간

의 강한 상관 관계를 나타낸다.

결론적으로, 두 강의 모두 United States 국가이고, MIT 강의이며, 학위 과정이 동일한 Bachelo라는 점에서 두 강의를 같이 들을 확률이 높고, 강한 양의 상관관계를 가져 이러한 결과가 나왔다고 해석할 수 있다.

4. 만일 하나의 규칙에 대한 효용성 지표를 Support × Confidence × Lift로 정의한다면 효용성이 가장 높은 규칙 1위~3위는 어떤 것들인가?

Support × Confidence × Lift의 값을 'Utility' 열을 만들어 계산하였고, 그 결과는 아래와 같다.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric	utility
(MITx_8.02x_India_Secondary)	(MITx_6.002x_India_Secondary)	0.007216	0.020414	0.002801	0.388109	19.011790	0.002653	1.600916	0.954287	0.020664
(MITx_8.02x_India_Bachelor's)	(MITx_6.002x_India_Bachelor's)	0.006474	0.022741	0.002497	0.385642	16.958041	0.002349	1.590700	0.947163	0.016327
(HarvardX_CS50x_India_Secondary)	(MITx_6.00x_India_Secondary)	0.009123	0.020432	0.002681	0.293926	14.385551	0.002495	1.387344	0.939052	0.011338

Lift를 기준으로 1위부터 3위까지, 각각 조건절과 결과절을 순서대로 나타내면 아래와 같다.

1. (MITx_8.02x_India_Secondary) (MITx_6.002x_India_Secondary)

2. (MITx_8.02x_India_Bachelor's) (MITx_6.002x_India_Bachelor's)

3. (HarvardX_CS50x_India_Secondary) (MITx_6.00x_India_Secondary)

1위는 confidence가 가장 높았던 규칙과 동일하고, 2위는 1위에서 학위 과정을 Secondary에서 Bachelor로 바꾼 것과 동일하다. 마지막으로 3위는 위에서 보았던 규칙이 아닌 새로운 규칙인 것을 알 수 있다.

1위 규칙을 먼저 살펴보면, confidence를 포함하여, lift 값도 이 중에서 가장 높은 수치를 보이는 것을 보아 (MITx_8.02x_India_Secondary)를 수강한 사용자들은 (MITx_6.002x_India_Secondary)도 수강할 확률이 높고, 결론적으로 매우 높은 상관관계를 보이는 것을 알 수 있다.

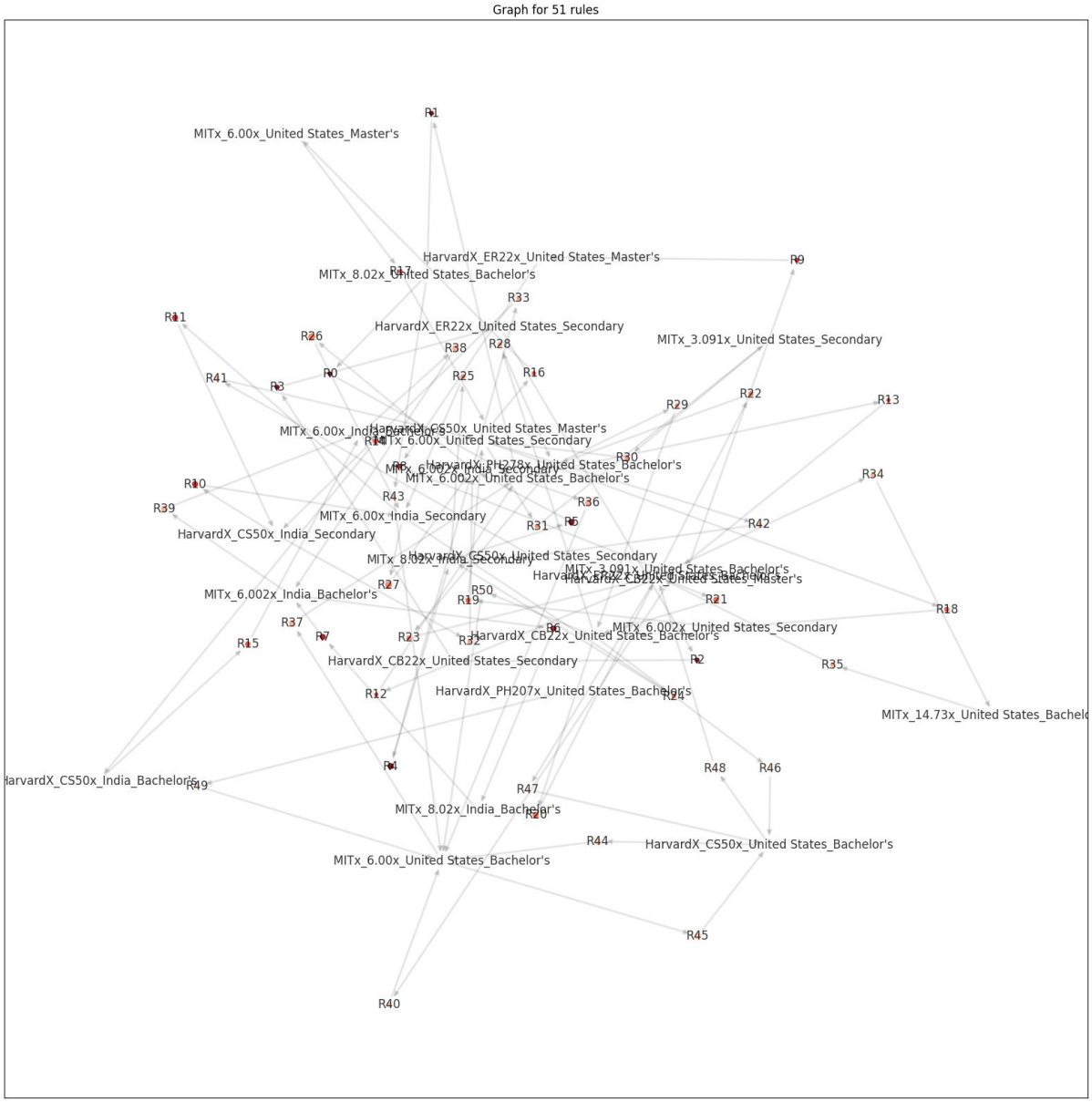
다음으로 2위는 1위보다 support, confidence, lift 모든 수치에서 낮은 수치를 보이고 둘의 차이가 학위 과정이라는 점을 고려했을 때, 학위 과정이 Secondary인 경우보다 Bachelor인 경우가 관련성이 덜 강하다고 해석할 수 있다.

마지막으로 3위는 앞선 규칙들과 달리 새로운 규칙이지만, 모든 수치에서 준수한 수치들을 보이는 것을 보아, 높은 상관관계를 보이고, 규칙으로써의 효용 가치가 높다고 할 수 있다. 다만, 1위에 비해 utility 값이 절반인 것을 알 수 있는데, 이를 통해 1위 규칙이 다른 규칙들에 비해 훨씬 더 효과적인 규칙이라고 결론을 내릴 수 있다.

[Extra Question] 이 외 수업 및 실습 시간에 다루지 않은 연관규칙분석 시각화 및 해석을 시도해 보시오.

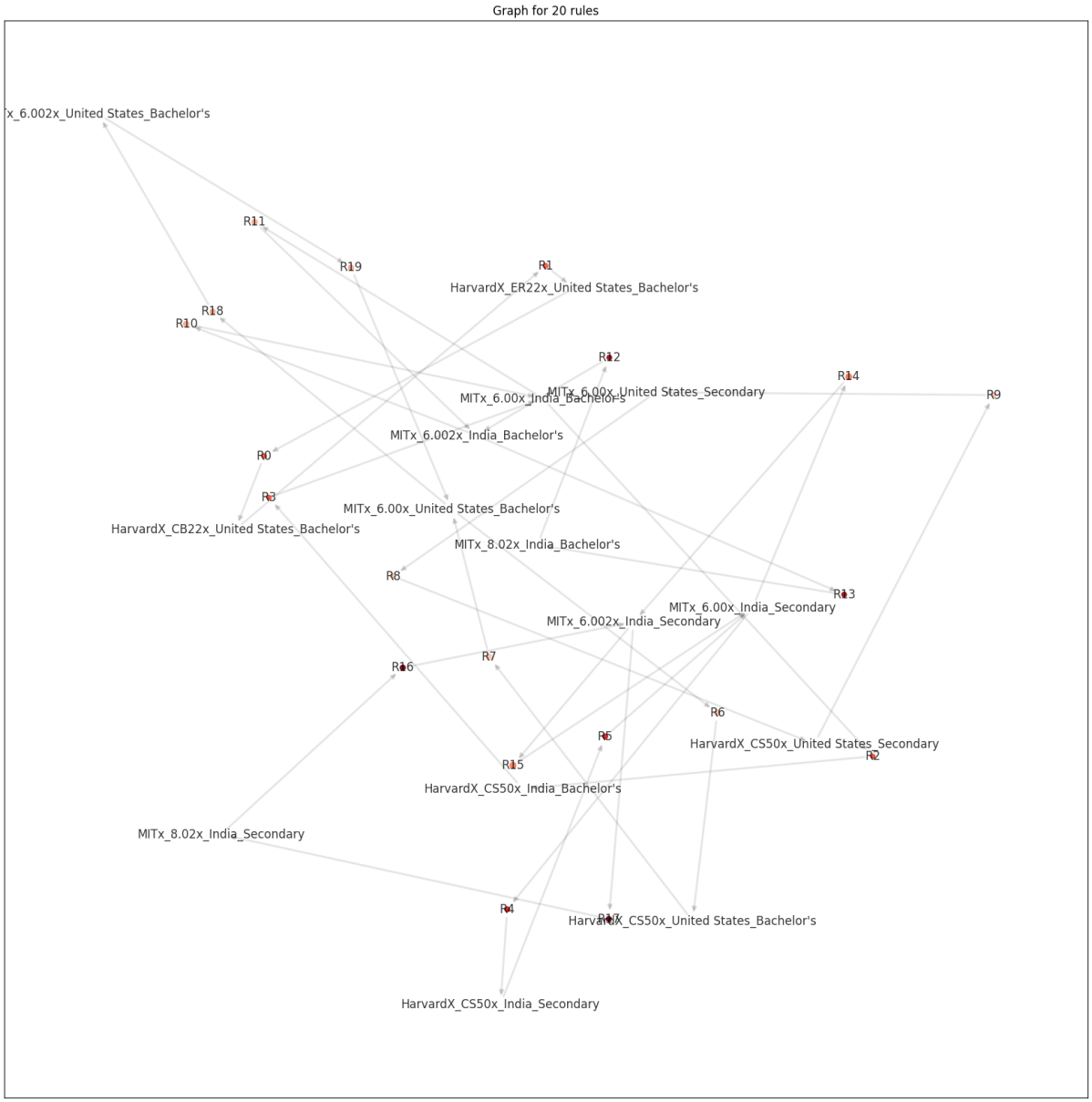
1. 연관 규칙 네트워크 그래프 (Association Rules Network Graph)

이 그래프는 연관 규칙 분석의 결과를 시각적으로 나타내어 규칙 간의 연결과 강도를 보여준다. 해당 그래프를 그린 결과는 아래와 같다.



해당 그래프에는 규칙 노드, 조건절 노드, 결과절 노드가 존재한다. 이때, 엣지를 통해, 조건절 노드, 규칙 노드, 결과절 노드가 연결되고, 결론적으로 각 규칙이 어떤 조건절과 어떤 결과절을 가지고 있는지 그래프로 파악이 가능하다. 또한, 노드의 색은 lift 값이 클수록 더 진하게 표현되고, support이 클수록 더 큰 크기로 표현되기 때문에 해석에 용이하다.

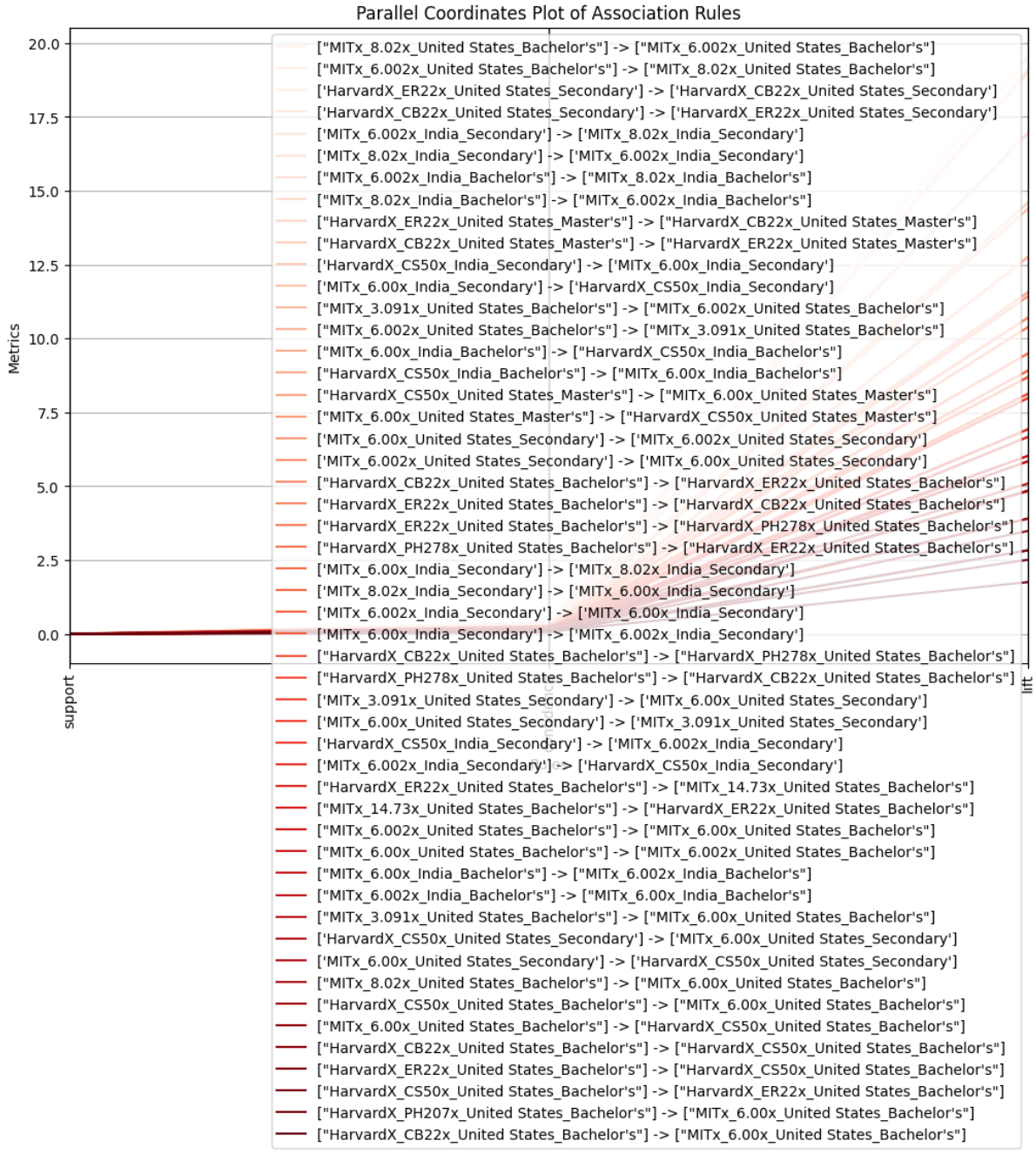
다만 위의 그래프는 support가 0.001, confidence가 0.05일 때로, rule이 너무 많아 보기 힘들어 rule의 개수를 줄이고자 min_support=0.002, min_confidence는 0.05로 설정하여 규칙의 개수를 줄여보았다. 결론적으로 20개의 규칙들이 만들어졌고, 이를 연관 규칙 네트워크 그래프 (Association Rules Network Graph)로 나타낸 결과는 아래와 같다.



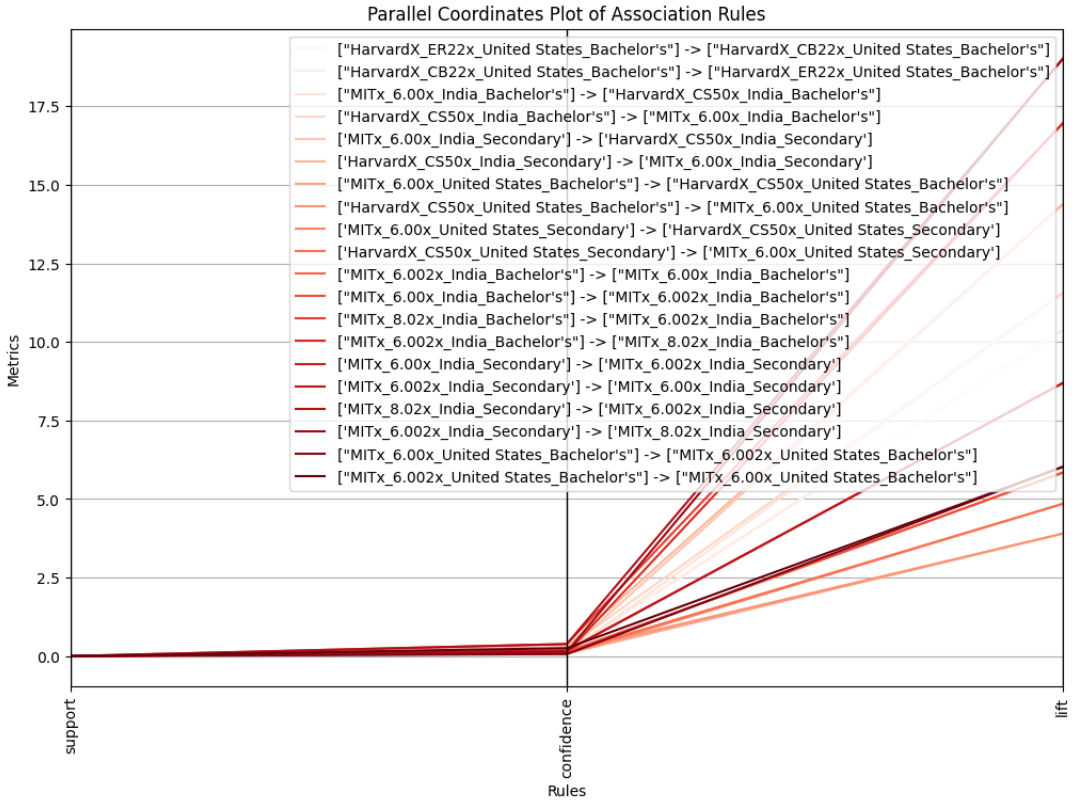
(MITx_8.02x_India_Secondary) (MITx_6.002x_India_Secondary)를 연결한 R16노드를 보면 lift 값이 크기 때문에 노드가 진한 색으로 표시되어 있는 것을 알 수 있다. 이처럼 조건절, 결과절의 연결 및 각 규칙들의 support, lift 등을 한눈에 파악하는 것이 가능하다.

2. Parallel Coordinates Plot

이 그래프는 각 규칙을 하나의 선으로 나타내고, 각 규칙이 가지는 지표들(support, confidence, lift)의 값에 따라 선의 위치와 두께를 조정하여 규칙들 간의 패턴을 시각적으로 비교할 수 있다.



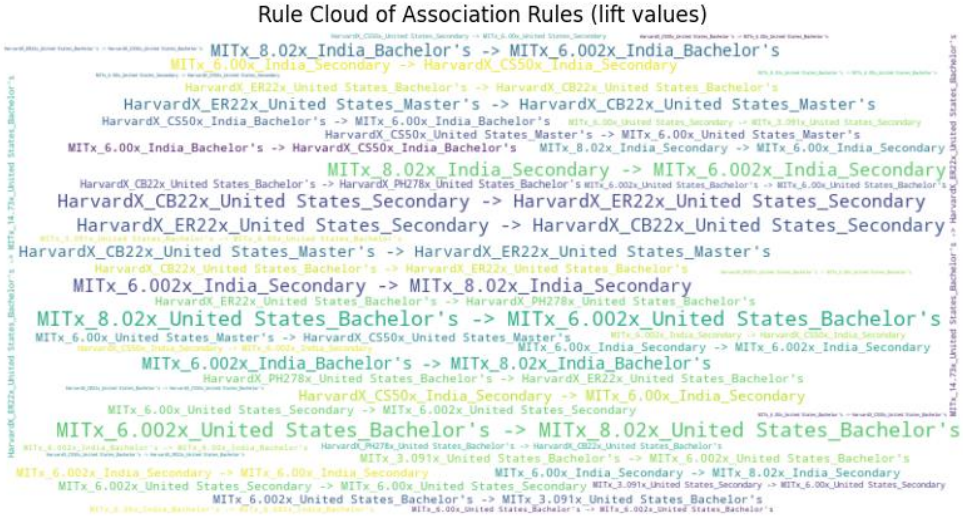
이것도 규칙이 너무 많기 때문에, min_support=0.002, min_confidence는 0.05로 설정하여 규칙의 개수를 줄여보았고, 이것을 Parallel Coordinates Plot으로 나타낸 결과는 아래와 같다.



해당 병렬 좌표계 플롯을 통해 규칙들의 지지도, 신뢰도, lift 등의 패턴을 비교하고 해석할 수 있지만, 위와 같이 rule의 개수를 조금 줄였음에도 색 구분 및, 그래프를 보기 힘든 것을 알 수 있다. 따라서 해당 plot은 규칙의 개수가 10개 이하일 때, 사용하는 것이 좋아 보인다.

3. Rule Cloud

Word cloud와 동일하게 각 규칙의 antecedents와 consequents를 조합하여 lift 지표를 기준으로 cloud를 생성해보았고, 그 결과는 아래와 같다.

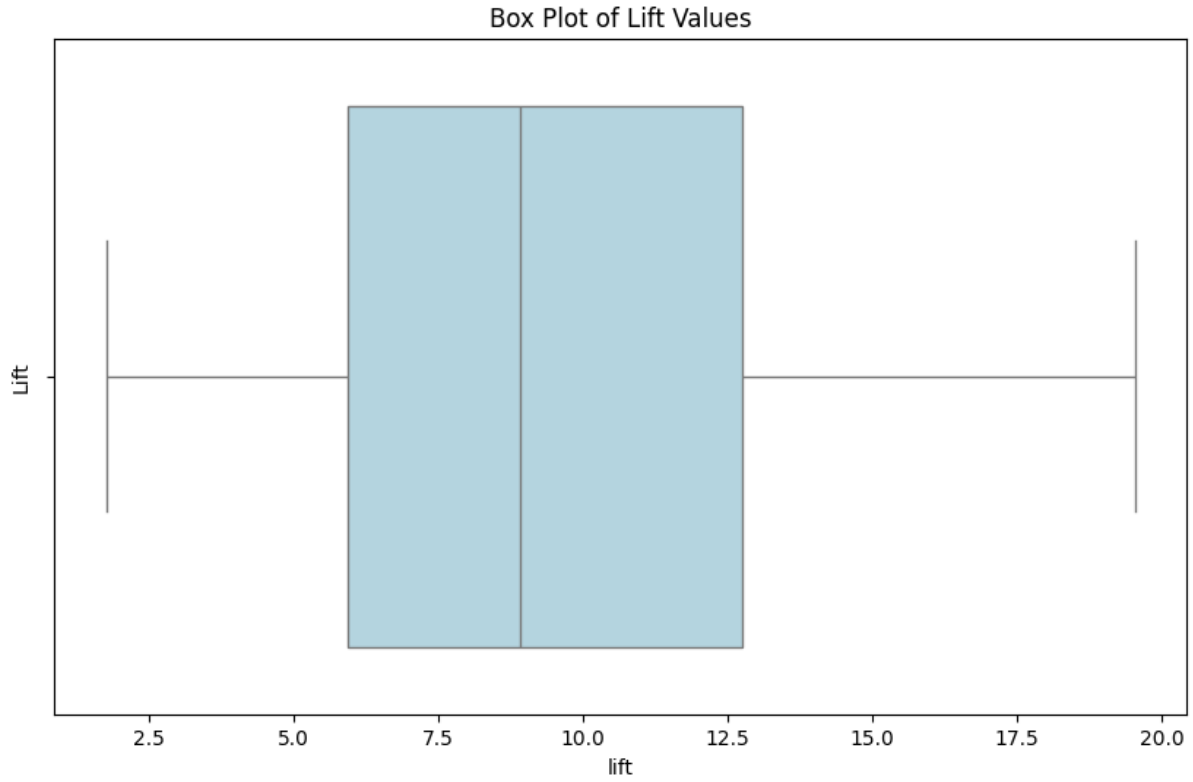


Word cloud와 동일하게, 여기서는 높은 lift 값을 가진 글자의 크기가 크기 때문에, 시각적으로 부각된다는 것을 알 수 있다.

MITx_8.02x_United States_Bachelor's -> MITx_6.002x_United States_Bachelor's 규칙의 글자 크기가 가장 큰 것을 알 수 있는데, 이는 위에서 가장 높은 lift 값을 가지는 규칙과 동일한 것을 알 수 있다. 결론적으로, Rule cloud를 통해 연관 규칙 분석 결과의 패턴을 쉽게 파악하고, 해석하는 것이 가능하다.

4. Box Plot

마지막으로 연관 규칙 분석 결과에서 lift 값의 분포를 시각화하는 박스 플롯을 생성해보았다. 이를 통해, lift 값의 중앙값, 사분위수 범위, 이상치 등을 시각적으로 표현하여 규칙들의 lift 값이 어떠한 분포를 가지는지 파악할 수 있다.



전체적으로 박스 plot이 왼쪽에 치우쳐져 있는 것을 보아, 규칙들의 lift 값들이 정규성을 띈다고 어려워 보인다. 이는 lift 값이 비교적 작은 값들이 많이 존재하거나, 다른 lift 값들에 비해 소수의 lift 값이 높은 수치를 보이기 때문에, 이러한 결과가 나왔다고 해석이 가능하다.

[Part 2: Clustering]

Dataset: Kaggle Clustering 데이터셋 중 1개 선택 Kaggle 사이트의 Datasets 항목에서 "clustering" 을 키워드로 검색하면 총 1,438개의 데이터셋이 아래와 같이 검색됩니다 (2024-05-26 기준).

https://www.kaggle.com/datasets?search=clustering

[Q1] 데이터셋 선정하기

이 중에서 군집화 후 각 군집에 대한 속성 분석이 유의미할(또는 재미있을) 것으로 판단되는 데이터셋 하나를 선정하고 본인이 해당 데이터셋을 선정한 이유를 설명하시오.

Dataset : **Help NGO To Identify The Nations In Dire Need of The Aid (Country-data.csv)**

링크: **https://www.kaggle.com/datasets/vipulgohel/clustering-pca-assignment/data**

자금 지원 프로그램을 통해 국제 인도주의 NGO가 1000만 달러를 모금했고, NGO는 이 자금을 전략적이고 효과적으로 사용할 방법을 결정해야 한다. 이 결정을 내릴 때, 중요한 점은 원조가 절실한 국가를 찾아 지원을 해야 한다는 것이다. 즉, NGO가 기부와 지원을 결정할 때는 특정 국가들에게 우선순위를 줘야 할 필요가 있고, 클러스터링을 통해 비슷한 개발 수준을 가진 국가들의 그룹을 형성하고, 이 그룹들 중 어떤 국가들이 가장 심각한 개발 도움이 필요한지를 판단할 수 있어야 한다.

결정적으로 국가들을 다양한 우선순위로 분류한다면, 향후 국제 인도적 NGO의 자원 배분에 큰 도움이 될 것이다. 따라서, 해당 데이터셋의 사회경제적 특성을 포함하고 있는 여러 변수들을 토대로, 비슷한 특성을 가진 국가들끼리 묶고, 국가들 간의 우선순위를 정해, NGO가 데이터 기반의 전략을 마련하여 지원을 결정할 수 있게 만들고자 한다.

해당 데이터셋은 총 10개의 변수가 존재한다.

Country(나라 이름), child_mort(아동 사망률), exports(수출), health(보건 지출), imports(수입), income(소득 per capita), inflation(물가 상승률), life_expec (기대 수명), total_fer (출산율), gdpp (국내 총생산)을 의미한다.

이러한 변수들은 각각 국가의 발전 수준과 경제적 특성을 종합적으로 평가할 수 있어, 이를 기반으로 한 클러스터링 분석은 국가 지원 계획을 수립하는 데 효과적일 수 있다.

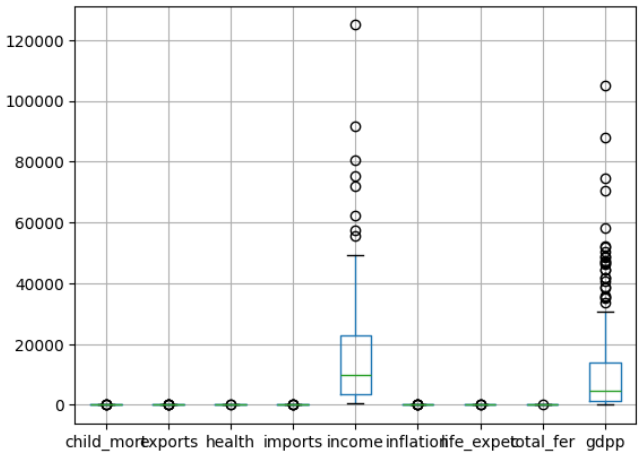
다음으로, 클러스터링을 진행하기 전, EDA를 진행하였다.

해당 데이터는 총 167개의 행과 10개의 열로 이루어진 데이터셋이다. 결측치 확인 결과, 결측치는 없는 것으로 나타났다.

다음으로 describe() 함수를 통해 각 변수들의 통계를 확인해 보았고, 그 결과는 아래와 같다.

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689	2.947964	12964.155689
std	40.328931	27.412010	2.746837	24.209589	19278.067698	10.570704	8.893172	1.513848	18328.704809
min	2.600000	0.109000	1.810000	0.065900	609.000000	-4.210000	32.100000	1.150000	231.000000
25%	8.250000	23.800000	4.920000	30.200000	3355.000000	1.810000	65.300000	1.795000	1330.000000
50%	19.300000	35.000000	6.320000	43.300000	9960.000000	5.390000	73.100000	2.410000	4660.000000
75%	62.100000	51.350000	8.600000	58.750000	22800.000000	10.750000	76.800000	3.880000	14050.000000
max	208.000000	200.000000	17.900000	174.000000	125000.000000	104.000000	82.800000	7.490000	105000.000000

위의 표를 확인해 보았을 때, 변수들마다 스케일 차이가 큰 것을 알 수 있다. 특히, income과 gdpp의 평균을 보았을 때, 10000이 넘는 것을 알 수 있는데, 다른 변수들의 평균이 100이 안 넘는 것을 확인할 수 있다. 이는 아래 boxplot을 통해서 쉽게 확인이 가능하다.



Box plot을 보았을 때, 이상치들도 존재하는 것을 볼 수 있는데, 해당 데이터셋에서 클러스터링을 통해 모든 나라들의 지원 우선순위를 정하는 것이 중요하다고 생각했기 때문에, 이상치 제거는 하지 않았다.

클러스터링을 진행할 때, 데이터 정규화는 중요하다. 그 이유로, 클러스터링 알고리즘은 거리를 척도로 사용하는 경우가 많기 때문에, 스케일이 큰 변수가 더 큰 영향을 미치게 된다. 그러나 변수의 스케일이 작다고 중요한 변수가 아니라고 할 수 없기 때문에, 변수들을 standardization하여 각 변수의 스케일을 맞추는 것이 중요하다.

데이터 스케일링을 진행하기 전, 카테고리형 변수인 country는 학습에 사용하지 않으므로 제거하였다. 그 후, StandardScaler를 진행하였고, 이렇게 전처리를 진행한 데이터로 학습을 진행하였다.

[K-Means Clustering]

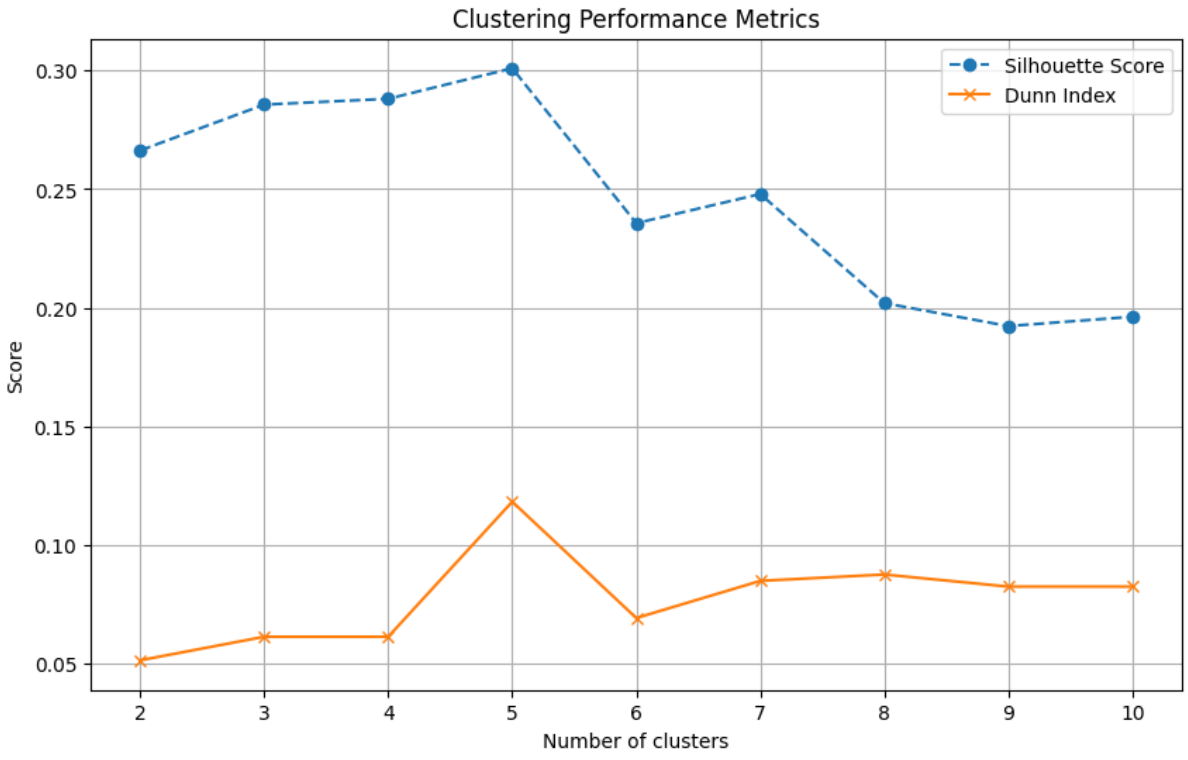
[Q2] K-Means Clustering의 군집 수를 2개부터 K(데이터의 규모에 따라서 본인이 적절히 선정할 것)개까지 증가시켜(증가 폭 역시 적절히 설정) 가면서 군집화 타당성 지표 값들을 산출하시오. 총 소요 시간은 얼마인가? Silhouette index 기준으로 가장 최적의 군집 수는 몇 개로 판별이 되었는가?

나라의 수가 총 167개로 많은 편이 아니므로, 너무 많은 군집 수는 좋지 않다고 생각하였다. 많은 군집이 형성되면, 각 군집의 특성을 이해하고 해석하는 것이 어려워지고, 결론적으로 NGO의 자원 배분을 위해 전략적 계획을 수립하는 데 있어 어려움이 있을 것이라 판단하였기에, 최대 군집 수를 10으로 설정하였다.

Random state=42로 설정하고 군집화 타당성 지표 값들을 산출한 결과는 다음과 같다. 이때, 총 소요 시간은 1초이다. 이때, 타당성 지표는 Silhouette index와 Dunn index를 사용하였다.

```
For n_clusters = 2, Silhouette score: 0.2662961111870726, Dunn index: 0.051534136217100665
For n_clusters = 3, Silhouette score: 0.285600988953231, Dunn index: 0.06145789081960917
For n_clusters = 4, Silhouette score: 0.2880471307804802, Dunn index: 0.06145789081960917
For n_clusters = 5, Silhouette score: 0.30088229124112015, Dunn index: 0.11834577178440615
For n_clusters = 6, Silhouette score: 0.23565028812238528, Dunn index: 0.06933778252544413
For n_clusters = 7, Silhouette score: 0.2479313491087983, Dunn index: 0.08498414259686386
For n_clusters = 8, Silhouette score: 0.20198237114728412, Dunn index: 0.08768306614308385
For n_clusters = 9, Silhouette score: 0.19230727869997855, Dunn index: 0.08257018679295938
For n_clusters = 10, Silhouette score: 0.19624516158796698, Dunn index: 0.08257018679295938
```

추가적으로, 군집 수마다 타당성 지표 값들을 산출한 결과를 그래프로 그려 시각화 하였고, 그래프는 아래와 같다.



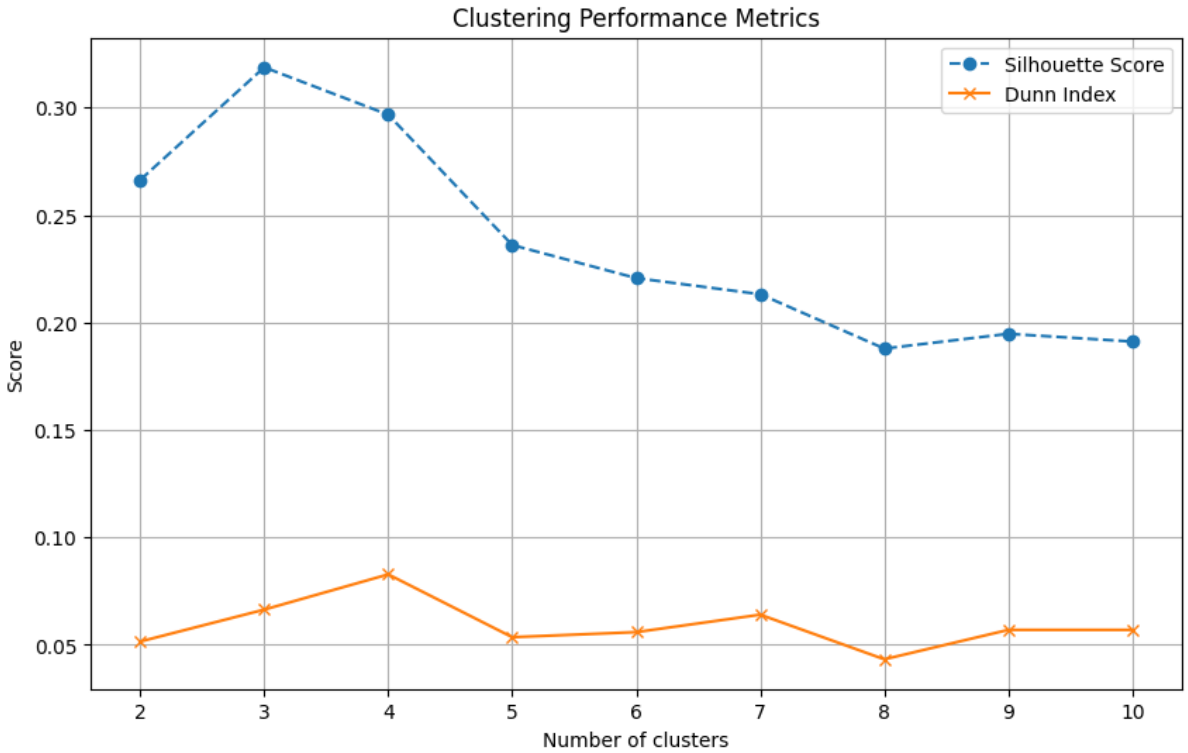
Silhouette score는 클러스터가 얼마나 명확하게 구분되는지를 측정한다. 값은 -1에서 1 사이에 있으며, 값이 클수록 클러스터가 더 명확하게 구분되는데, 클러스터의 수가 5일 때, 약 0.30으로 가장 높은 값을 가지기 때문에, 이때 클러스터가 비교적 명확하게 구분된다고 할 수 있다.

Dunn Index는 클러스터 간의 분리를 측정하는 지표로, 값이 클수록 클러스터 간의 분리가 잘 된 것을 의미한다. Dunn index 또한, 클러스터 수가 5일 때 가장 높은 값을 기록하기 때문에 이때, 클러스터 간의 분리가 가장 잘 된다고 할 수 있다.

결론적으로 Silhouette index 기준으로 가장 최적의 군집 수는 5개인 것을 알 수 있고, 이때 Dunn index도 가장 높은 값을 보이는 것을 확인할 수 있다.

추가로, K-Means는 초기에 클러스터 중심을 랜덤하게 선택하고 이는 클러스터링 결과에 큰 영향을 주기 때문에, Random seed를 다르게 하여, 시도를 해보았다. 그 결과 Random seed를 10000으로 설정했을 시, 위와 다른 결과를 보이는 것을 알 수 있다. 군집 타당성 지표 값들과 그래프를 나타낸 결과는 아래와 같다.

```
For n_clusters = 2, Silhouette score: 0.2662961111870726, Dunn index: 0.051534136217100665
For n_clusters = 3, Silhouette score: 0.31871826290334204, Dunn index: 0.06638488827968289
For n_clusters = 4, Silhouette score: 0.29666713905612757, Dunn index: 0.08278146297901762
For n_clusters = 5, Silhouette score: 0.2360161153399637, Dunn index: 0.05355759323421662
For n_clusters = 6, Silhouette score: 0.22063389907786157, Dunn index: 0.055917966725051074
For n_clusters = 7, Silhouette score: 0.21312831614447098, Dunn index: 0.06403632468160264
For n_clusters = 8, Silhouette score: 0.1878431956136106, Dunn index: 0.04326746060894352
For n_clusters = 9, Silhouette score: 0.19472033464318778, Dunn index: 0.05695828676918532
For n_clusters = 10, Silhouette score: 0.19111707566549604, Dunn index: 0.05695828676918532
```



이 경우, Silhouette index 기준으로 가장 최적의 군집 수는 3개인 것을 알 수 있고, Dunn index 기준으로 가장 최적의 군집 수는 4인 것을 보아, 두 지표의 결과가 일치하지 않는 것을 알 수 있다. 또한, 최적의 군집 수일 때, 실루엣 계수가 약 0.32로, 이는 이전의 결과인 0.30보다 더 높은 수치임을 알 수 있다. 결론적으로 두 개의 결과를 비교한다면, 군집 수가 3개인 경우가 최적의 군집 수라고 결론 내릴 수 있다.

다만, Random seed 값에 결과가 다르게 나온다는 점을 고려하여, 이후 문제에서도 최적의 클러스터링 개수를 3과 5 두 가지를 사용하고자 한다.

[Q3] [Q2]에서 선택된 군집의 수를 사용하여 K-Means Clustering을 10회 반복하고 회차마다 각 군집의 Centroid와 Size를 확인해보시오. 10회 반복 시 몇 가지 경우의 군집화 결과물이 도출되었으며 각 경우의 군집화는 몇번 반복되어 발생하는지 확인해보시오

먼저, 최적의 군집 수가 3이라고 하였을 때, 군집화를 10회 반복하였고, 모든 centroid 점이 일치할 때, 같은 군집화 결과물이 도출된다고 하였다. Centroid 점에 따른 반복 횟수는 아래와 같다.

1. ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.643094825842432, 0.12853723865046973, 0.18791487378522995, -0.04680774224700341, 0.3998113000295032, -0.20717568248727736, 0.6523307973065732, -0.6554779197255397, 0.34944888296528603), (0.9453546031190024, -0.3992146066340747, -0.26493145273743535, -0.13456808526638003, -0.6720663529160315, 0.3147277050272104, -0.975062198181368, 0.971212771011883, -0.6010218356184638)): **2번 반복 발생하였습니다**.

이때 각 군집의 개체 수는 {3, 68, 96}이다.

2. ((-0.8274486635067486, 0.6450798457714426, 0.7274112195080666, 0.19063895239524611, 1.4842426811375093, -0.4849206439106801, 1.0795785301840388, -0.7918768655237743, 1.6159953561392424), (-0.3932819644433522, -0.030583746766141402, -0.20617902848599914, 0.01956246789353044, -0.25093020946788924, -0.005783145520533908, 0.22676243863119686, -0.40257861878153445, -0.35583225404494906), (1.4135644628526947, -0.4576149383529719, -0.1878979433887638, -0.1898972116682761, -0.7078386334824861, 0.398988748790009, -1.297031040197961, 1.4028739638681755, -0.6127613105143801)): **2번 반복 발생하였습니다**.

이때 각 군집의 개체 수는 {86, 45, 36}이다.

3. ((-0.8274486635067486, 0.6450798457714426, 0.7274112195080666, 0.19063895239524611, 1.4842426811375093, -0.4849206439106801, 1.0795785301840388, -0.7918768655237743, 1.6159953561392424), (-0.40645336634961077, -0.031652588260933805, -0.22447089689610297,

0.02416160623738873, -0.2517704138171778, -0.01716742039350983, 0.25473362072412625, -0.424342785902827, -0.35448141229501007), (1.3602177587151119, -0.4375331283798608, -0.15598401197909897, -0.1892037704291374, -0.6868940800065406, 0.40211077646466614, -1.282179813350041, 1.3649438547807105, -0.6042424295368484)): **5번 반복 발생하였습니다**.

이때 각 군집의 개체 수는 {84, 47, 36}이다.

4. 군집화 결과 ((-0.8485887325812405, 4.267903898351802, 0.2106226351818973, 3.821303352043224, 2.2010787776652245, -0.639133456620417, 1.1976884442407236, -0.9278847197250164, 2.3194939995201773), (-0.5640705552208806, 0.0649294622392574, 0.09711628131751672, -0.052769664897187374, 0.26091966566741487, -0.18281818581378007, 0.5421184939463389, -0.5598170839479989, 0.21567565286238816), (1.201445286100017, -0.44720235143493015, -0.21163268896920018, -0.1765429617477671, -0.689714049414984, 0.4163651125959941, -1.182993881798403, 1.198733352393185, -0.6071596696311297)): **1번 반복 발생하였습니다**.

이때 각 군집의 개체 수는 {54, 109, 4}이다.

결론적으로 각 centroid 점이 ((-0.8274486635067486, 0.6450798457714426, 0.7274112195080666, 0.19063895239524611, 1.4842426811375093, -0.4849206439106801, 1.0795785301840388, -0.7918768655237743, 1.6159953561392424), (-0.40645336634961077, -0.031652588260933805, -0.22447089689610297, 0.02416160623738873, -0.2517704138171778, -0.01716742039350983, 0.25473362072412625, -0.424342785902827, -0.35448141229501007), (1.3602177587151119, -0.4375331283798608, -0.15598401197909897, -0.1892037704291374, -0.6868940800065406, 0.40211077646466614, -1.282179813350041, 1.3649438547807105, -0.6042424295368484))이고, 군집 별 개체 수는 {84, 47, 36}인 경우가 5번 발생으로 가장 빈번하게 발생한 결과물이라고 할 수 있다.

다음으로, 최적의 군집 수가 5라고 하였을 때, 군집화를 10회 반복하였고, 모든 centroid 점이 일치할 때, 같은 군집화 결과물이 도출된다고 하였다. Centroid 점에 따른 반복 횟수는 아래와 같다.

각 경우의 군집화가 몇 번 반복되었는지 확인:

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -

1.0388627117187226, 2.4407973524626976), (-0.8467483030382849, 0.03450531781204013, 1.267768203334536, -0.28529709631174605, 1.1656090362309401, -0.6179701735890187, 1.1549433071746917, -0.7701303322999331, 1.6957576310214473), (-0.6516959323776419, 0.7440763284112493, -1.2001746993407107, -0.42080139106011394, 2.092686222746719, 0.8190051049432983, 0.689220564893243, -0.4153246737839352, 0.8957756991948028), (-0.4268185739730903, -0.003684011948167733, -0.14350600314534598, 0.09887784977122148, -0.27207977953923934, -0.11265842040562576, 0.2674267332358052, -0.4596117291174267, -0.35028713380311505), (1.3454188284243331, -0.4596718596234268, -0.18959690487796174, -0.22567743349440947, -0.6871351496611285, 0.392267026907683, -1.2642923504925934, 1.3290818811320615, -0.6053076907046262)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8262939538514195, 0.14381032829211632, 0.8852371345643021, -0.3402098349412096, 1.5206212871882532, -0.46696693389384347, 1.1122948421110705, -0.7449721657567114, 1.7423584443548759), (-0.5683515554774239, 0.4077728769137211, 0.14641710547071682, 0.6364966935235645, -0.14452454889870572, -0.36907218568790845, 0.37066841472656564, -0.6161525093785163, -0.24545354173071368), (-0.28233885623430177, -0.3707102560691125, -0.5208335369495526, -0.5421771672816511, -0.2519921540152398, 0.2918544781517273, 0.1862039123483793, -0.25292503840210045, -0.380595813535248), (1.3905284557844804, -0.44568533632301405, -0.17203067943513872, -0.18179731209474823, -0.6967993827304647, 0.3926628132875535, -1.3018257224431407, 1.371397791879824, -0.6073088589642681)): **2번 반복 발생하였습니다**.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8262939538514195, 0.14381032829211632, 0.8852371345643021, -0.3402098349412096, 1.5206212871882532, -0.46696693389384347, 1.1122948421110705, -0.7449721657567114, 1.7423584443548759), (-0.5131050277034386, 0.15865279327192996, -0.023404073052405872, 0.2930585498566637, -0.1735740049619199, -0.3038572706444718, 0.36116622627702777, -0.5078583459603877, -0.2758774853913782), (0.15322756313346164, -0.43426275949187937, -0.7521455260616682, -0.7823879165753652, -0.3745615580752518, 1.1634507199237634, -0.18479063851314628, 0.21450890758641875, -0.4697572479514426), (1.490628632098034, -0.4522929011407872, -0.035132878267358364, -0.04342336933938175, -0.7120297818466877, 0.04773087812575591, -1.431217621322253, 1.389402008621178, -0.6171032057033602)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8319564723535154, 0.49661053158653745, 0.7837939190786868,

0.009856654347920952, 1.5814527374735887, -0.48385318097083424, 1.119092092280589, -0.7804661930468095, 1.7729448698148722), (-0.5183615097836259, 0.6857354854592481, -0.08910460067361396, 0.8774591989116377, -0.1588005709985056, -0.38431028504698284, 0.3292516225825503, -0.5620070280442735, -0.2823017913298246), (-0.33307502201308165, -0.3666432013935434, -0.28405409215053196, -0.47368938877613914, -0.23308078452243597, 0.2149247522883841, 0.17500905577289147, -0.3425662998179122, -0.3473191724333019), (1.3539932878136505, -0.3765894100448807, -0.5884036624627951, -0.35411853679281974, -0.6912572565860765, 0.5634601250962541, -1.1534552143950638, 1.5293904522533244, -0.6057096464982112), (1.5879907582751667, -0.7598652260479941, 1.4059409376058771, 0.6069431681707818, -0.8109300698484435, -0.20085689491835623, -1.6743561873859498, 1.0177085548392721, -0.6616298407366524)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8286092947500543, 0.17262065695945344, 0.8591897727478232, -0.2963727632897516, 1.4622751233406923, -0.47818851097005205, 1.107649215328545, -0.7636814619194032, 1.6619021388317752), (-0.40661978339100663, 0.009928983620761618, -0.18502990752434323, 0.05286737937492944, -0.21620377301362298, -0.09993352265620589, 0.24599239751762086, -0.4204778807090163, -0.3341390300555079), (0.8733942468836007, 0.4232016921752434, -0.9418818377142946, -0.3812700983861226, -0.2804820531500197, 2.7878021226635132, -0.7125906034002257, 1.029965825596786, -0.35435567895925346), (1.430616814196643, -0.621412132355482, -0.05432172894528207, -0.16159721470932062, -0.7726486209818777, 0.05769418569115515, -1.3489887987587064, 1.3940738720422678, -0.6480644916881572)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8286092947500543, 0.17262065695945344, 0.8591897727478232, -0.2963727632897516, 1.4622751233406923, -0.47818851097005205, 1.107649215328545, -0.7636814619194032, 1.6619021388317752), (-0.4199500586618423, 0.009825957446776291, -0.2026089511069951, 0.05819605518489291, -0.21621699998340776, -0.11339916787504499, 0.27408697650090236, -0.4424071579095285, -0.332293651761699), (0.48406464142837946, -0.2784130097370622, -0.6118775866632266, -0.6762874747173507, -0.3828263035040724, 5.242571527664682, -0.35967066126532926, 0.4651375379870276, -0.3723459694614431), (1.3401923833846037, -0.4344696826787041, -0.14551761915465125, -0.16675635699342323, -0.6882599344907979, 0.21238018344691353, -1.2853984872508488, 1.3529614739366775, -0.6047273459611434)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8229408546599821, 0.18330800824238536, 0.8298940979945345, -0.26125205807349156, 1.3983777122162335, -0.499856229544024, 1.0743309411642785, -0.7682503319878009, 1.5954361099168415), (-0.3991871695176221, -0.0009218664200527092, -0.207778328575796, 0.037485831374689464, -0.2323445457432807, -0.005131907207700545, 0.23341762938792351, -0.4072267771873798, -0.35404699542045875), (1.3938412682830172, -0.45486864989206066, -0.17768112205887895, -0.16644522128637348, -0.7097425848404812, 0.200563082733389, -1.3007335453408038, 1.3912089772832197, -0.6134618795256325), (2.2813850239184923, -0.5784516306330678, -0.6374380819036966, -1.2217847884719917, -0.6240647737306978, 9.129718055281284, -1.134120813912878, 1.9161333736062311, -0.5819362740192721)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8262939538514195, 0.14381032829211632, 0.8852371345643021, -0.3402098349412096, 1.5206212871882532, -0.46696693389384347, 1.1122948421110705, -0.7449721657567114, 1.7423584443548759), (-0.5210037527056567, 0.20998006247564777, -0.01780673574461066, 0.37040873412870096, -0.16194028642005273, -0.3153633707671791, 0.35601613401926824, -0.5138132175652615, -0.26907380919275203), (0.12924308990357863, -0.4408536345006818, -0.6057917544631148, -0.7519367890044371, -0.37403928038534556, 0.9272010399573078, -0.21367803540215907, 0.12300140678255012, -0.4580214027765603), (1.5684495179542075, -0.4746194500498231, -0.05132027675691913, -0.03214688541937526, -0.7401283007300871, 0.04877354952865299, -1.4363819414510475, 1.5127319221118383, -0.6322817257668446)): 1번 반복 발생하였습니다.

군집화 결과 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8278774681532903, 0.1681120867099368, 0.8873776686484794, -0.30560645417905974, 1.4889172658047993, -0.47955551791646617, 1.108725200642736, -0.7507654437741759, 1.7019696710196859), (-0.5113779836972068, 0.14247740410520898, -0.058831932501057324, 0.25591120782227067, -0.15425551632426562, -0.26616393870100413, 0.3467331430185266, -0.5004893057825169, -0.2769116314562155), (0.35990635800587495, -0.35505953276014124, -0.6829744263657106, -0.6082941793497371, -0.4472140315387214, 1.0575214428748774, -0.3890829600380664, 0.3468519521394776, -0.4972905690046753), (1.6166160965893168, -0.5672149364122574, 0.05257749928834616, -0.0649473655581073, -0.7901311189929638, -0.06143628729117213, -1.512129133017546,

1.5525829366658475, -0.6576715796408045)): 1번 반복 발생하였습니다.

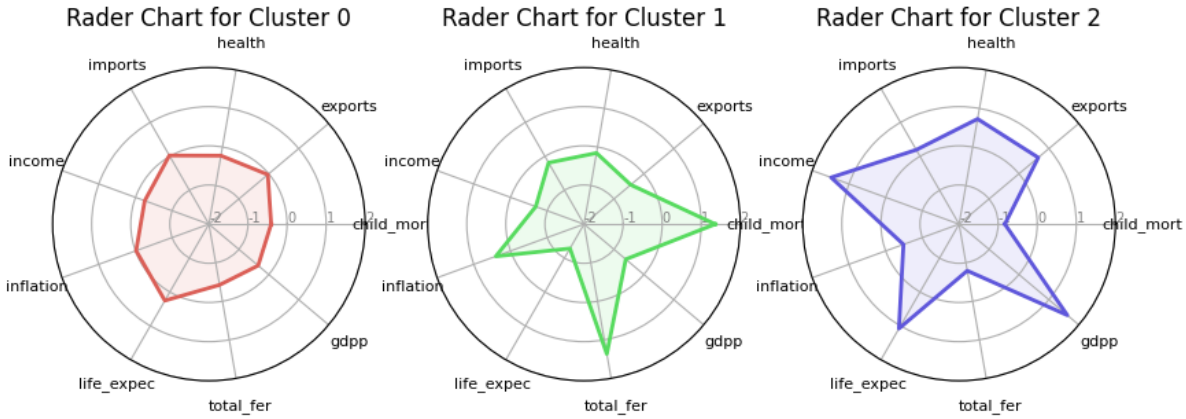
이때, 유일하게 centroid가 ((-0.8490032437395639, 4.93567278022401, -0.008163032412125565, 4.548057684608741, 2.439542398485936, -0.5042061410238958, 1.226824311634017, -1.0388627117187226, 2.4407973524626976), (-0.8262939538514195, 0.14381032829211632, 0.8852371345643021, -0.3402098349412096, 1.5206212871882532, -0.46696693389384347, 1.1122948421110705, -0.7449721657567114, 1.7423584443548759), (-0.5683515554774239, 0.4077728769137211, 0.14641710547071682, 0.6364966935235645, -0.14452454889870572, -0.36907218568790845, 0.37066841472656564, -0.6161525093785163, -0.24545354173071368), (-0.28233885623430177, -0.3707102560691125, -0.5208335369495526, -0.5421771672816511, -0.2519921540152398, 0.2918544781517273, 0.1862039123483793, -0.25292503840210045, -0.380595813535248), (1.3905284557844804, -0.44568533632301405, -0.17203067943513872, -0.18179731209474823, -0.6967993827304647, 0.3926628132875535, -1.3018257224431407, 1.371397791879824, -0.6073088589642681))인 경우가 2번 발생하였고, 이때의 군집 별 개체 수는 {46, 45, 28, 45, 3}인 것을 알 수 있다.

[Q4] [Q3]에서 가장 빈번하게 발생한 군집화 결과물에 대해서 각 군집별 변수들의 평균값을 이용한 Radar Chart를 도시해보시오. Radar Chart상으로 판단할 때, 군집의 속성이 가장 상이할 것으로 예상되는 두 군집(군집 A와 군집 B로 명명)과, 가장 유사할 것으로 예상되는 두 군집(군집 X와 군집 Y로 명명)을 각각 선택하고 선택 이유를 설명하시오.

최적의 군집 수가 3일 때, 가장 빈번하게 발생한 군집화 결과물에 대해 각 군집별 변수들의 평균값을 구한 결과는 아래와 같다.

clusterID	0	1	2
child_mort	-0.406453	1.360218	-0.827449
exports	-0.031653	-0.437533	0.645080
health	-0.224471	-0.155984	0.727411
imports	0.024162	-0.189204	0.190639
income	-0.251770	-0.686894	1.484243
inflation	-0.017167	0.402111	-0.484921
life_expec	0.254734	-1.282180	1.079579
total_fer	-0.424343	1.364944	-0.791877
gdpp	-0.354481	-0.604242	1.615995

다음으로, 평균값을 이용하여 Radar Chart를 그려보았고, 그 결과는 아래와 같다.



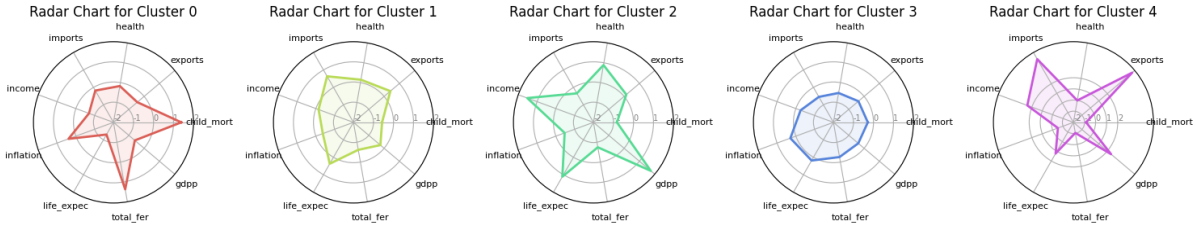
해당 결과를 보았을 때, 군집의 속성이 가장 상이할 것으로 보이는 두 군집은 Cluster1과 Cluster2이다. 그 이유로, Cluster0는 모두 비슷한 수치를 보이고 있는 반면, Cluster1과 Cluster2는 변수마다 대조적인 모습을 보이기 때문이다. Cluster1은 Child_mort, total_fer, inflation에서 높은 수치를 보이고, 나머지는 낮은 수치를 보이는 반면, Cluster2는 Child_mort, total_fer, inflation에서 낮은 수치를 보이고, 나머지는 높은 수치를 보이기 때문이다. 즉, Cluster1과 Cluster2는 정반대의 속성을 가지고 있다고 해석이 가능하다.

반면, 군집의 속성이 가장 유사할 것을 보이는 두 군집은 Cluster0와 Cluster1이다. Cluster0는 Cluster1과 Cluster2의 중간 값을 보이고, 또한 두 군집과는 확실히 다른 특징을 보이고 있긴 하지만, Cluster1이 높은 수치를 보이는 Child_mort, total_fer, inflation을 제외한 나머지 변수에서는 두 군집 간 비슷한 수치를 보이기 때문이다. Cluster2는 child_mort와 total_fer을 제외한 나머지 변수에서 Cluster0와 평균값 차이를 보이기 때문에, Cluster0과 Cluster1이 유사할 것으로 예상한다.

추가로, 최적의 군집 수가 5일 때, 가장 빈번하게 발생한 군집화 결과물에 대해 각 군집별 변수들의 평균값을 구한 결과는 아래와 같다.

clusterID	0	1	2	3	4
child_mort	1.390528	-0.568352	-0.826294	-0.282339	-0.849003
exports	-0.445685	0.407773	0.143810	-0.370710	4.935673
health	-0.172031	0.146417	0.885237	-0.520834	-0.008163
imports	-0.181797	0.636497	-0.340210	-0.542177	4.548058
income	-0.696799	-0.144525	1.520621	-0.251992	2.439542
inflation	0.392663	-0.369072	-0.466967	0.291854	-0.504206
life_expec	-1.301826	0.370668	1.112295	0.186204	1.226824
total_fer	1.371398	-0.616153	-0.744972	-0.252925	-1.038863
gdpp	-0.607309	-0.245454	1.742358	-0.380596	2.440797

다음으로, 평균값을 이용하여 Radar Chart를 그려보았고, 그 결과는 아래와 같다.



해당 결과를 보았을 때, 군집의 속성이 가장 상이할 것으로 보이는 두 군집은 Cluster0와 Cluster4이다. 그 이유는 Cluster0와 Cluster4가 각 변수마다 상반된 모습을 보이기 때문이다. Cluster0는 Child_mort, total_fer, inflation에서 매우 높은 수치를 보이고, 나머지는 낮은 수치를 보이는 반면, Cluster2는 Child_mort, total_fer, inflation에서 매우 낮은 수치를 보이고, 특히, income, imports, exports에서 매우 높은 수치를 보이기에, 가장 상이할 것으로 예상된다.

반면, 군집의 속성이 가장 유사할 것을 보이는 두 군집은 Cluster1과 Cluster3이다. Cluster1과 Cluster3가 다른 군집들에 비해, 특출난 부분 없이 모든 변수에서 비슷한 수치를 보이는 동글동글한 모양을 하고 있기 때문에 가장 유사할 것으로 예상된다.

[Q5] [Q4]에서 선택된 군집 A와 군집 B에 대해 각 변수별 평균값 차이에 대한 통계적 검정을 수행하시오. 전체 변수 중에서 유의수준 0.05에서 값의 차이가 나타나는 변수의 비중은 얼마인가? 또한 [Q4]에서 선택된 군집 X와 군집 Y에 대해서도 각 변수별 평균값 차이에 대한 통계적 검정을 수행하시오. 이 경우, 전체 변수 중에서 유의수준 0.05에서 값의 차이가 나타나는 변수의 비중은 얼마인가?

최적의 군집 수가 3일 때, 가장 상이할 것으로 선택된 클러스터는 Cluster1과 Cluster2이다. 두 군집에 대해 각 변수별 평균값 차이에 대한 통계적 검정을 수행하였고, 그 결과는 다음과 같다.

	two_sided	greater	less
0	1.305756e-22	6.528778e-23	1.000000e+00
1	2.645762e-04	9.998677e-01	1.322881e-04
2	4.577720e-04	9.997711e-01	2.288860e-04
3	1.753041e-01	9.123480e-01	8.765204e-02
4	2.666084e-14	1.000000e+00	1.333042e-14
5	2.224988e-04	1.112494e-04	9.998888e-01
6	3.203735e-28	1.000000e+00	1.601868e-28
7	3.733201e-28	1.866601e-28	1.000000e+00
8	6.359976e-15	1.000000e+00	3.179988e-15

0~8까지의 수는 각각 (child_mort, exports, health, imports, income, inflation, life_expec, total_fer, gdpp) 변수를 의미한다. 이때, 유의수준 0.05에서 값의 차이가 나타나는 변수는 imports를 제외한 모든 변수임을 알 수 있다. 즉, 9개의 변수 중, 8개의 변수에서 값의 차이가 나타나기 때문에, 그

비중은 약 89%라고 할 수 있다. 또한, 예상과 동일하게 평균 값의 차이가 큰 것을 알 수 있다.

추가로, 각 변수 평균 값에서 cluster1이 cluster2보다 큰지 작은지 나타내기 위해 단측 검정도 진행하였고, 그 결과를 표로 나타낸 것은 다음과 같다.

child_mort	Cluster1 > cluster2
exports	Cluster1 < cluster2
health	Cluster1 < cluster2
imports	Cluster1 = cluster2
income	Cluster1 < cluster2
inflation	Cluster1 > cluster2
life_expec	Cluster1 < cluster2
total_fer	Cluster1 > cluster2
gdpp	Cluster1 < cluster2

다음으로, 가장 유사할 것으로 선택된 클러스터는 Cluster0와 Cluster1이다. 두 군집에 대해 각 변수별 평균값 차이에 대한 통계적 검정을 수행하였고, 그 결과는 다음과 같다.

	two_sided	greater	less
0	1.107357e-19	1.000000e+00	5.536783e-20
1	1.362481e-03	6.812403e-04	9.993188e-01
2	6.810343e-01	6.594828e-01	3.405172e-01
3	1.318685e-01	6.593427e-02	9.340657e-01
4	2.550947e-10	1.275473e-10	1.000000e+00
5	7.277226e-02	9.636139e-01	3.638613e-02
6	3.983001e-20	1.991500e-20	1.000000e+00
7	7.517848e-25	1.000000e+00	3.758924e-25
8	1.278278e-09	6.391391e-10	1.000000e+00

이때, 유의수준 0.05에서 값의 차이가 나타나는 변수는 child_mort, exports, income, life_expec, total_fer, gdpp인 것을 알 수 있다. 총 9개의 변수 중, 6개의 변수에서 값의 차이가 나타나기 때문에, 그 비중은 약 66.7%라고 할 수 있다. 이는 가장 상이할 것으로 예측한 군집보다 차이가 크지 않다는 것이고, 나온 결과가 예상과 일치하는 것을 알 수 있다.

추가로, 각 변수 평균 값에서 cluster0이 cluster1보다 큰지 작은지 나타내기 위해 단측 검정도 진행하였고, 그 결과를 표로 나타낸 것은 다음과 같다.

child_mort	Cluster0 < cluster1
exports	Cluster0 > cluster1
health	Cluster0 = cluster1
imports	Cluster0 = cluster1
income	Cluster0 > cluster1

inflation	Cluster0 = cluster1
life_expec	Cluster0 > cluster1
total_fer	Cluster0 < cluster1
gdpp	Cluster0 > cluster1

최적의 군집 수가 5일 때도, 위와 동일하게 통계적 검정을 수행하였다. 먼저, 가장 상이할 것으로 예상한 클러스터는 Cluster0와 Cluster4이고, 통계적 검정 수행결과는 다음과 같다.

	two_sided	greater	less
0	1.716174e-22	8.580871e-23	1.000000e+00
1	6.651174e-03	9.966744e-01	3.325587e-03
2	7.885678e-01	6.057161e-01	3.942839e-01
3	3.965380e-03	9.980173e-01	1.982690e-03
4	8.429493e-02	9.578525e-01	4.214747e-02
5	1.373109e-03	6.865545e-04	9.993134e-01
6	1.946896e-11	1.000000e+00	9.734480e-12
7	1.696330e-08	8.481651e-09	1.000000e+00
8	1.540569e-01	9.229715e-01	7.702846e-02

이때, 유의수준 0.05에서 값의 차이가 나타나는 변수는 child_mort, exports, imports, inflation, life_expec, total_fer 인 것을 알 수 있다. 총 9개의 변수 중, 6개의 변수에서 값의 차이가 나타나기 때문에, 그 비중은 약 66.7%라고 할 수 있다. 이는 가장 상이할 것으로 예상했던 것에 비해 높지 않은 수치라고 할 수 있다.

추가로, 각 변수 평균 값에서 cluster0이 cluster4보다 큰지 작은지 나타내기 위해 단측 검정도 진행하였고, 그 결과를 표로 나타낸 것은 다음과 같다.

child_mort	Cluster0 > cluster4
exports	Cluster0 < cluster4
health	Cluster0 = cluster4
imports	Cluster0 < cluster4
income	Cluster0 = cluster4
inflation	Cluster0 > cluster4
life_expec	Cluster0 < cluster4
total_fer	Cluster0 > cluster4
gdpp	Cluster0 > cluster4

반면, 군집의 속성이 가장 유사할 것을 예상한 두 군집은 Cluster1과 Cluster3이었고, 통계적 검정 수행결과는 다음과 같다.

	two_sided	greater	less
0	5.162359e-05	9.999742e-01	0.000026
1	1.678078e-08	8.390389e-09	1.000000
2	3.125464e-05	1.562732e-05	0.999984
3	1.250258e-15	6.251291e-16	1.000000
4	2.820457e-01	1.410229e-01	0.858977
5	1.432981e-05	9.999928e-01	0.000007
6	5.555686e-02	2.777843e-02	0.972222
7	2.892196e-04	9.998554e-01	0.000145
8	4.536801e-02	2.268401e-02	0.977316

이때, 유의수준 0.05에서 값의 차이가 나타나는 변수는 child_mort, exports, health, imports, inflation, total_fer, gdpp 인 것을 알 수 있다. 총 9개의 변수 중, 7개의 변수에서 값의 차이가 나타나기 때문에, 그 비중은 약 77.8%라고 할 수 있다. 이는 가장 상이할 것으로 예상했던 군집들의 비중보다 높은 수치로, 예상한 것과 다른 결과라고 할 수 있다.

추가로, 각 변수 평균 값에서 cluster0이 cluster4보다 큰지 작은지 나타내기 위해 단측 검정도 진행하였고, 그 결과를 표로 나타낸 것은 다음과 같다.

child_mort	Cluster1 < cluster3
exports	Cluster1 > cluster3
health	Cluster1 > cluster3
imports	Cluster1 > cluster3
income	Cluster1 = cluster3
inflation	Cluster1 < cluster3
life_expec	Cluster1 = cluster3
total_fer	Cluster1 < cluster3
gdpp	Cluster1 > cluster3

군집의 개수가 3개일 때, [Q4]에서 예상했던 군집 조합과, '값의 차이가 나타나는 변수의 비중'과 일치하는 것을 알 수 있었는데, 군집의 개수가 5개인 경우는 예상과 다른 것을 알 수 있다. 이는 다음과 같이 해석할 수 있다. 모든 변수에서 차이가 나지는 않지만, 특정 변수에서 평균 값의 차이가 매우 크게 난다면 Radar 차트를 보았을 때, 형태의 차이가 크게 나타날 수 있다. 결과적으로, Radar 차트를 보고 두 군집 속성이 다르다는 결론을 내릴 수 있지만, 몇몇 변수의 큰 차이로 인해 선택한 결과이기 때문에, 이는 '값의 차이가 나타나는 변수의 비중'과 일치하지 않을 수 있다. 따라서, 보다 정확하게 체크하려면, '값의 차이가 나타나는 변수의 비중'과 더불어 '값의 차이가 얼마나 큰지'도 같이 고려하여야 한다고 생각한다.

[Hierarchical Clustering]

[Q6] 두 객체 사이의 유사도를 측정하는 지표를 본인의 기준에 따라 정의하고(유클리드 거리, 상관계수 등) “single”과 “complete” 두 가지 linkage에 대해 군집 수를 2개부터 K(데이터의 규모에 따라서 본인이 적절히 선정할 것)개까지 증가시켜(증가 폭 역시 적절히 설정) 가면서 군집화 타당성 지표 값들을 산출하시오. Silhouette index 기준으로 가장 최적의 군집 수는 몇 개로 판별이 되었는가?

유사도를 측정하는 지표로 유클리드 거리를 선택하였다.

상관계수는 두 변수 간의 선형적 관계의 강도와 방향을 나타내는 통계적 측도인데, 데이터가 비선형적인 관계를 가지는 경우 상관계수는 이를 잘 반영하지 못할 수 있다. 또한, 해당 데이터는 이상치들이 존재하는데 상관계수는 이러한 이상치에 민감하게 반응할 수 있어, 계층적 군집화에 사용한다면 군집화의 정확성을 저하시킬 수 있다.

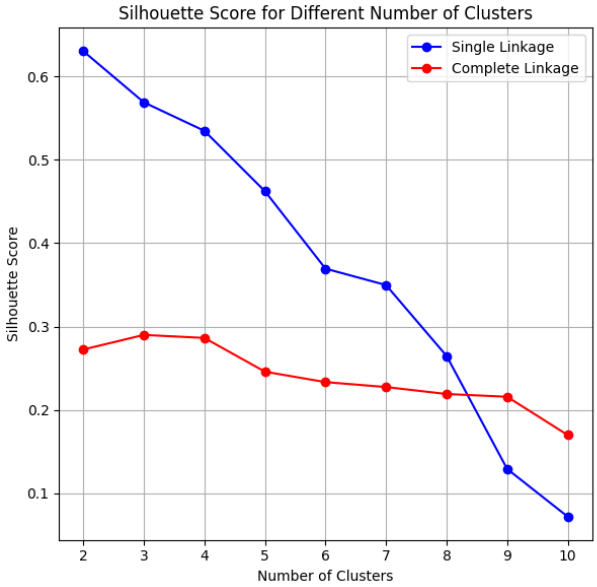
반면, 유클리드 거리는 데이터 포인트 간의 절대적인 차이를 계산하는 방법으로, 숫자형 데이터에 적합하고, 거리 행렬을 계산하여 군집화 하기 때문에, 거리 기반의 명확한 군집 구조를 도출하는 데 더 적합하다고 할 수 있다. 따라서, 해당 데이터 셋에서 유사도를 측정하는 지표로 유클리드 거리를 사용하고자 한다.

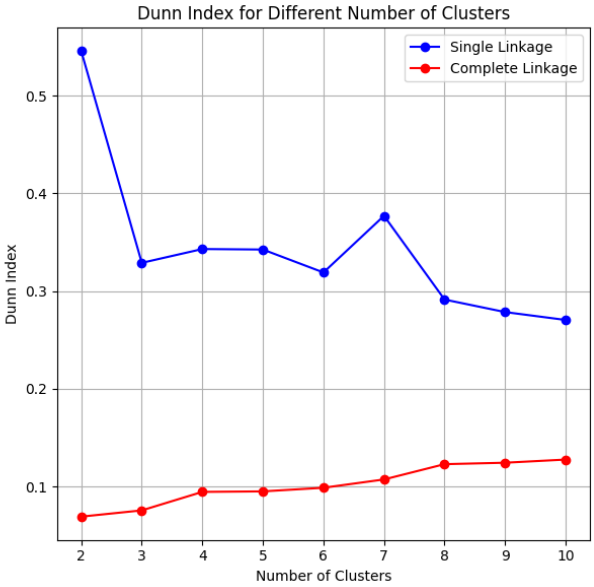
[Q2]의 답변과 동일하게, 나라의 수가 총 167개로 많은 편이 아니고, 많은 군집이 형성되면, 각 군집의 특성을 이해하고 해석하는 것이 어려워지기 때문에, 최대 군집 수를 10으로 설정하였다.

Single과 complete 두 가지 linkage에 대해 군집 수를 2개부터 10개까지 증가시켜 가면서 silhouette index와 Dunn index 군집화 타당성 지표 값들을 산출하였고, 그 결과는 아래와 같다.

```
For n_clusters = 2: Single Linkage - Silhouette score: 0.6303375987432929, Dunn index: 0.5456674553929378
For n_clusters = 2: Complete Linkage - Silhouette score: 0.2725501366968193, Dunn index: 0.06925274202135723
For n_clusters = 3: Single Linkage - Silhouette score: 0.5689127494518488, Dunn index: 0.3289725337757121
For n_clusters = 3: Complete Linkage - Silhouette score: 0.29005247526122313, Dunn index: 0.07565540410359314
For n_clusters = 4: Single Linkage - Silhouette score: 0.5347720470494712, Dunn index: 0.3431083003403182
For n_clusters = 4: Complete Linkage - Silhouette score: 0.2864378723295819, Dunn index: 0.09457489318145268
For n_clusters = 5: Single Linkage - Silhouette score: 0.4623542746724967, Dunn index: 0.3425921605316398
For n_clusters = 5: Complete Linkage - Silhouette score: 0.245986430937637, Dunn index: 0.09510240793094364
For n_clusters = 6: Single Linkage - Silhouette score: 0.369496226532075, Dunn index: 0.3191278589942445
For n_clusters = 6: Complete Linkage - Silhouette score: 0.23346469538642686, Dunn index: 0.09879934339472674
For n_clusters = 7: Single Linkage - Silhouette score: 0.3498136530713941, Dunn index: 0.37710049789376776
For n_clusters = 7: Complete Linkage - Silhouette score: 0.22755293122128029, Dunn index: 0.10730406988871383
For n_clusters = 8: Single Linkage - Silhouette score: 0.2648562258572818, Dunn index: 0.2914065369483415
For n_clusters = 8: Complete Linkage - Silhouette score: 0.21919769094860928, Dunn index: 0.1229589227054121
For n_clusters = 9: Single Linkage - Silhouette score: 0.1293594496695663, Dunn index: 0.2785960449665999
For n_clusters = 9: Complete Linkage - Silhouette score: 0.21589758443753077, Dunn index: 0.1244353490775003
For n_clusters = 10: Single Linkage - Silhouette score: 0.07217193522634151, Dunn index: 0.270523281365448
For n_clusters = 10: Complete Linkage - Silhouette score: 0.17003097905821968, Dunn index: 0.12765884003598751
```

이것을 눈으로 보기 쉽게, 그래프로 시각화 하였고, 그 결과는 아래와 같다.



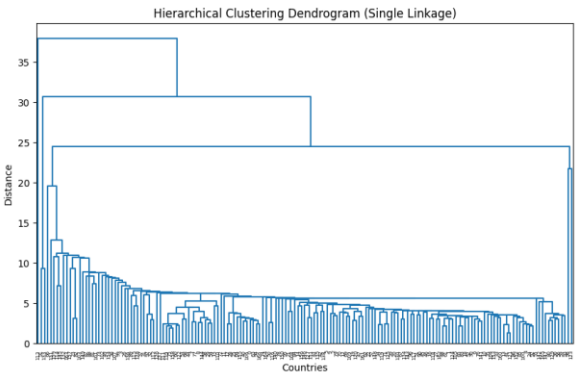


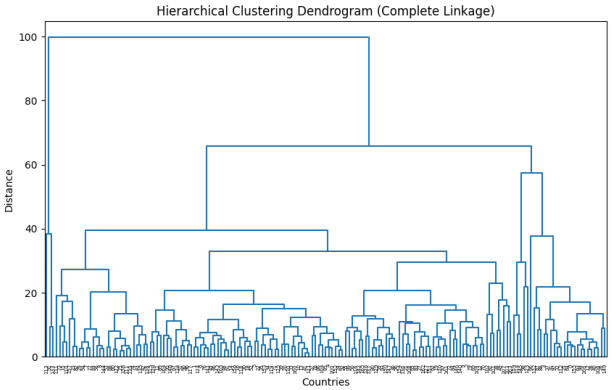
이때, silhouette index는 single linkage, complete linkage 모두 클러스터의 수가 많아질수록, 값이 감소하는 모습을 보이는데, dunn index는 군집의 수가 많아질수록, single linkage는 감소하고, complete linkage는 증가하는 모습을 보인다.

군집의 수가 증가하면 각 지표의 값이 어떻게 변하는지는 군집화 방법과 데이터의 성격에 따라 달라질 수 있다. 이때, Single linkage는 군집 간 거리가 작게 형성되어 Dunn index가 감소하고, Complete linkage는 군집 간 거리가 크게 형성되어 Dunn index가 증가할 수 있다. 이러한 이유로 위와 같은 결과가 나올 수 있다.

다음으로 Silhouette index 기준으로 가장 최적의 군집 수를 판별하면, single linkage에서는 군집의 수가 2개일 때, 가장 높은 Silhouette index를 보이기 때문에 최적의 군집 수는 2개, complete linkage에서는 군집의 수가 3개일 때, 가장 높은 Silhouette index를 보이기 때문에 최적의 군집 수는 3개라고 할 수 있다.

추가로, single linkage를 통해 만든 덴드로그램과, complete linkage를 통해 만든 덴드로그램은 아래와 같다.



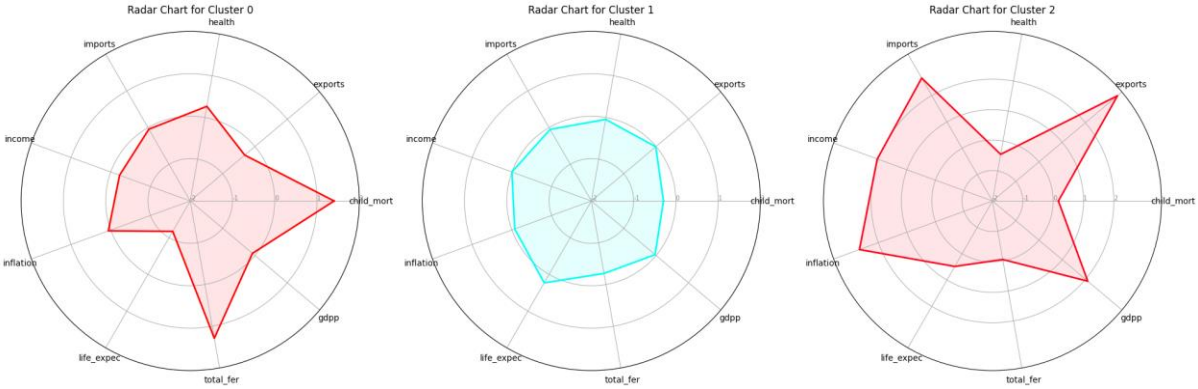


[Q7] [Q6]에서 찾은 최적의 군집 수에 대해서 각 군집들의 변수값의 평균값을 이용한 Radar Chart를 도시 해보시오. Radar Chart를 바탕으로 판단할 때, K-Means Clustering과 보다 유사한 결과물이 나오는 방식은 어떤 Linkage인지 본인의 생각을 바탕으로 서술해보시오.

먼저, complete linkage 기준, 최적의 군집의 수는 3개이다. 이때, 각 변수들의 평균 값을 구한 결과는 다음과 같다.

	Cluster 0	Cluster 1	Cluster 2
child_mort	1.404516	-0.292967	0.172352
exports	-0.315630	-0.009535	3.378155
health	0.267612	-0.045305	-0.443909
imports	-0.044428	-0.049565	2.661604
income	-0.222177	0.000785	2.038056
inflation	0.060458	-0.071212	2.664024
life_expec	-1.173192	0.230897	0.482450
total_fer	1.287416	-0.263965	-0.049447
gdpp	-0.077395	-0.030354	2.098412

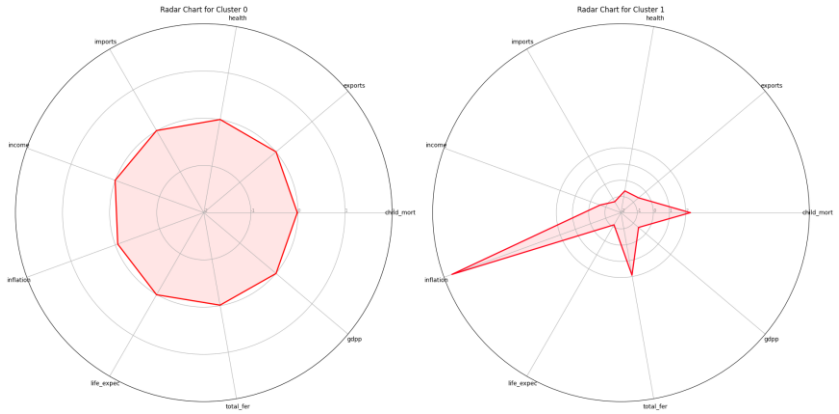
평균값을 이용하여 Radar Chart를 도시한 결과는 다음과 같다.



다음으로, single linkage 기준, 최적의 군집의 수는 2개이다. 이때, 각 변수들의 평균 값을 구한 결과는 다음과 같다.

	Cluster 0	Cluster 1
child_mort	-0.013743	2.281385
exports	0.003485	-0.578452
health	0.003840	-0.637438
imports	0.007360	-1.221785
income	0.003759	-0.624065
inflation	-0.054998	9.129718
life_expec	0.006832	-1.134121
total_fer	-0.011543	1.916133
gdpp	0.003506	-0.581936

평균값을 이용하여 Radar Chart를 도시한 결과는 다음과 같다.

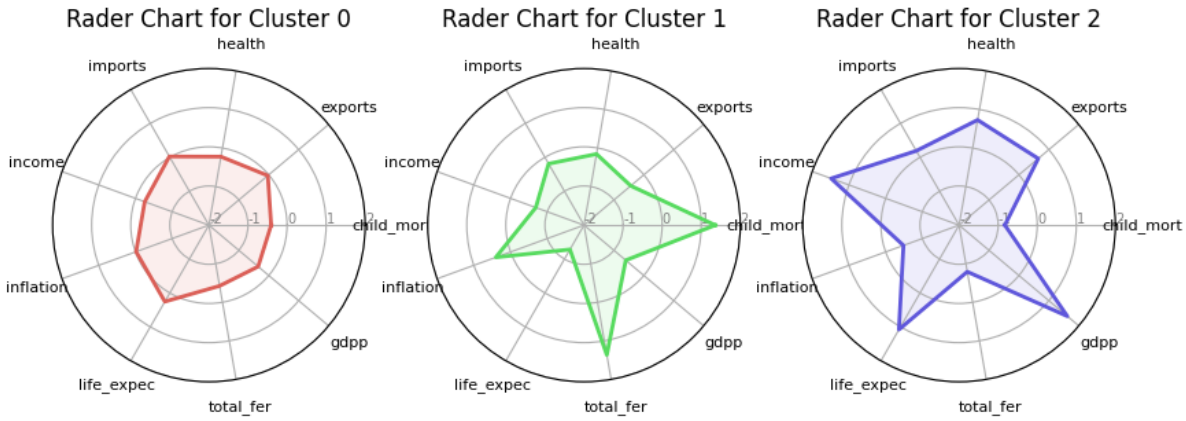


둘의 결과를 비교해보았을 때, complete linkage 기준으로 만든 cluster들은 cluster 0, 1, 2끼리 모두 서로 다른 특징을 가지고 있다. 먼저, cluster 0의 경우, child_mort, total_fer에서 높은 값을 보이고 life_expert에서 특히 낮은 수치를 보이는 것을 알 수 있다. 반면, cluster 2의 경우, exports, gdpp, inflation, Income, imports에서 높은 값을 보이고, child_mort, total_fer, health에서 낮은 수치를 보이는 것을 알 수 있다. Cluster 1은 모든 변수에서 비슷한 평균값을 보이는 것을 알 수 있다. 즉, 각각의 cluster마다 뚜렷한 특징을 보인다고 할 수 있다.

Single linkage 기준으로 만든 cluster들은 둘이 매우 큰 차이를 보이는 것으로 보인다. 그러나, single linkage의 cluster1과 complete linkage의 cluster0는 서로 비슷한 양상을 보이지만, complete linkage의 cluster0, 2처럼 분명히 다른 특징을 지닌 개체들이 존재하는데도 불구하고, single linkage의 cluster1은 극단적인 값을 가진 개체들만 모여, 군집을 형성했다고 해석할 수 있다.

즉, complete linkage에서는 서로 다른 성향을 보이는 개체들끼리 군집화가 잘 되었다고 해석이 가능하지만, single linkage에서는 극단적인 값을 가진 개체를 제외하고, 나머지는 모두 cluster0에 포함되어 두 클러스터 간 극단적인 차이를 보인다고 할 수 있다.

다음으로, K-Means Clustering의 결과와 비교하기 위해, K-Means Clustering Rader Chart를 다시 한번 살펴보고자 한다.



이는 complete linkage를 통해 만들어진 cluster와 상당히 유사하다고 할 수 있다. K-Means의 cluster0는 complete linkage의 cluster1과 유사하고, K-Means의 cluster1은 complete linkage의 cluster0와 유사하고, K-Means의 cluster2는 complete linkage의 cluster2와 유사하다. 반면, complete linkage를 기준으로 만들어진 cluster와는 다소 동떨어진 형태를 가지고 있다고 판단된다. 결과적으로, K-Means Clustering과 보다 유사한 결과물이 나오는 방식은 complete linkage라고 할 수 있다.

[DBSCAN]

[Q8] DBSCAN 알고리즘의 eps 옵션과 minPts 옵션을 조정해가면서 [Q2]에서 선정한 최적 개수의 군집이 찾아지는 eps 값과 minPts 값을 찾아보시오.

이때, eps 값이 너무 작은 경우, minPts 값이 너무 높은 경우에 클러스터 개수가 0으로 나타날 수 있다. 그 이유는 eps 값을 작게 설정하면, 데이터 포인트들 간의 거리가 eps보다 크게 되어, 반경 안에 들어오는 개체 수가 부족하여 클러스터를 형성하지 못할 수 있기 때문이다. 또한, minPts 값을 크게 설정하면, eps 반경 안에 포함된 개체 수가 충분하지 않아 클러스터를 형성하지 못하는 데이터 포인트가 많아질 수 있기 때문이다.

반면, eps 값이 너무 크거나, minPts 값이 너무 작은 경우, 클러스터의 개수가 줄어들 수 있다. Eps 값을 크게 설정하면, 더 멀리 떨어진 개체들도 하나의 클러스터로 간주될 가능성이 높고, 클러스터들이 크게 합쳐져 하나의 큰 클러스터가 형성될 수 있다. 또한 minPts 값을 작게 설정하면, eps 반경 내에 포함된 이웃의 수가 적어도 클러스터가 형성되기 시작하여, 더 많은 데이터들이 클러스터에 포함될 수 있다.

따라서, eps와 minPts를 너무 크거나 너무 작지 않게 조정하여 적절한 클러스터를 형성할 수 있게 하여야 한다.

[Q2]에서 선정한 최적 개수의 군집을 3개라고 하였을 때, 최적의 군집을 찾기 위해 사용한 eps, mitPts 후보는 다음과 같다.

1. eps_values = [0.5, 0.7, 0.9, 1.1, 1.3, 1.5]

2. minPts_values = [2, 3, 4, 5, 6]

위에서 선정한 후보들은 위에서 설명한 것을 고려하여 최대한 넓은 범위의 조합으로 시행해보고자 나타낸 결과이다.

각 조합에 따라 나타난 결과는 다음과 같다.

```
Eps: 0.5, MinPts: 2, 클러스터 개수: 3
Eps: 0.5, MinPts: 3, 클러스터 개수: 1
Eps: 0.5, MinPts: 4, 클러스터 개수: 0
Eps: 0.5, MinPts: 5, 클러스터 개수: 0
Eps: 0.5, MinPts: 6, 클러스터 개수: 0
Eps: 0.7, MinPts: 2, 클러스터 개수: 15
Eps: 0.7, MinPts: 3, 클러스터 개수: 8
Eps: 0.7, MinPts: 4, 클러스터 개수: 2
Eps: 0.7, MinPts: 5, 클러스터 개수: 1
Eps: 0.7, MinPts: 6, 클러스터 개수: 0
Eps: 0.9, MinPts: 2, 클러스터 개수: 14
Eps: 0.9, MinPts: 3, 클러스터 개수: 4
Eps: 0.9, MinPts: 4, 클러스터 개수: 3
Eps: 0.9, MinPts: 5, 클러스터 개수: 3
Eps: 0.9, MinPts: 6, 클러스터 개수: 4
Eps: 1.1, MinPts: 2, 클러스터 개수: 8
Eps: 1.1, MinPts: 3, 클러스터 개수: 4
Eps: 1.1, MinPts: 4, 클러스터 개수: 3
Eps: 1.1, MinPts: 5, 클러스터 개수: 4
Eps: 1.1, MinPts: 6, 클러스터 개수: 4
Eps: 1.3, MinPts: 2, 클러스터 개수: 4
Eps: 1.3, MinPts: 3, 클러스터 개수: 2
Eps: 1.3, MinPts: 4, 클러스터 개수: 2
Eps: 1.3, MinPts: 5, 클러스터 개수: 2
Eps: 1.3, MinPts: 6, 클러스터 개수: 2
Eps: 1.5, MinPts: 2, 클러스터 개수: 3
Eps: 1.5, MinPts: 3, 클러스터 개수: 2
Eps: 1.5, MinPts: 4, 클러스터 개수: 1
Eps: 1.5, MinPts: 5, 클러스터 개수: 1
Eps: 1.5, MinPts: 6, 클러스터 개수: 1
```

1. Eps: 0.5, MinPts: 2

2. Eps: 0.9, MinPts: 4

3. Eps: 0.9, MinPts: 5

4. Eps: 1.1, MinPts: 4

5. Eps: 1.5, MinPts: 2

위의 조합에서 최적 군집 개수인 3개의 클러스터가 만들어지는 것을 확인할 수 있다.

또한, 위의 결과에서 Eps: 0.5는 작은 편이기 때문에, MinPts 값이 4 이상만 되어도 클러스터의 개수가 0이 되어, 군집이 만들어지지 않는 것을 확인할 수 있다. 반면, Eps가 0.7, 0.9로 크지 않으면서, MinPts가 2인 경우에는 조그만한 군집들이 여러 개 만들어지는 것을 확인할 수 있다.

이처럼, DBSCAN 알고리즘을 적용할 때는 원하는 클러스터 개수와 형태를 얻을 수 있도록, eps와 minPts를 적절히 조정하는 것이 필수적이라고 할 수 있다.

[Q9] [Q8]에서 찾은 군집화 결과물에서 Noise로 판별된 객체의 수가 몇 개인지 확인해 보시오.

[Q8]에서 찾은 군집화 결과물에 대해 Noise로 판별된 객체의 수는 다음과 같다.

1. Eps: 0.5, MinPts: 2, 클러스터 개수: 3, Noise로 판별된 객체의 수: 158

2. Eps: 0.9, MinPts: 4, 클러스터 개수: 3, Noise로 판별된 객체의 수: 112

3. Eps: 0.9, MinPts: 5, 클러스터 개수: 3, Noise로 판별된 객체의 수: 119

4. Eps: 1.1, MinPts: 4, 클러스터 개수: 3, Noise로 판별된 객체의 수: 64

5. Eps: 1.5, MinPts: 2, 클러스터 개수: 3, Noise로 판별된 객체의 수: 23

전체 개체의 수가 167개인데, 1번의 경우는 노이즈로 판별된 객체가 158개로 너무 많기 때문에 클러스터링 결과가 지나치게 불안정하고 분석에 적합하지 않다고 할 수 있다. 이와 동일하게, 2,3,4의 경우도 노이즈로 판별된 데이터가 전체 데이터에 비해 많은 부분을 차지하므로, 클러스터링 결과의 신뢰성이 낮고, 적절한 클러스터링 결과라고 할 수 없다. 결론적으로 5번을 제외한 모든 군집화 결과는 데이터의 대다수가 노이즈로 처리되었기 때문에, 분석에 적절하지 않다고 평가할 수 있다.

그러나, 5번도 노이즈의 개수가 다소 많다고 판단하여 추가적으로 eps와 minPts를 설정하여 경우의 수를 찾아보았고, 그 결과 Eps: 1.7, MinPts: 2인 경우 Noise로 판별된 객체의 수가 18개로 가장 적은 것을 알 수 있었다.

또한, 지금까지 최적의 클러스터 개수를 3,5 두 가지의 경우를 열어놓고, 과제를 진행하였는데 eps_values = [0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1], minPts_values = [2, 3, 4, 5, 6, 7, 8]

위의 파라미터 조합에서는 DBSCAN의 결과 클러스터 개수가 5개가 나오는 경우가 없었다.

결과적으로 DBSCAN에서는 Eps: 1.7, MinPts: 2, 클러스터 개수: 3, Noise로 판별된 객체의 수: 18가 최적의 결과라고 할 수 있다.

[종합]

[Q10] 이 데이터셋에 가장 적합한 군집화 알고리즘은 무엇이라고 생각하는지 본인이 생각한 근거를 이용하여 서술하시오.

이 데이터셋에 가장 적합한 군집화 알고리즘은 Hierarchical Clustering이라고 생각한다. 가장 중요한 이유는 해당 데이터셋의 분석 목적이다. NGO가 자금을 전략적으로 효과적으로 사용하기 위해서, 원조가 절실한 국가를 찾아 지원해야 하는데, 이때 비슷한 개발 수준을 가진 국가들의 그룹을 형성하여, 어떤 국가 그룹이 가장 도움이 필요한지 판단하는 과정이 매우 중요하다고 할 수 있다.

DBSCAN의 경우, 해당 데이터셋에서 노이즈 처리가 되어 그룹화 되지 못한 나라가 있다면, 해당 나라들의 우선순위를 판단하기 어렵기 때문에, 적절하지 않다고 할 수 있다. 실제로, 위에서 Noise의 개수를 더 줄이기 위해 여러 파라미터 조합을 만들어 DBSCAN을 시도해보았지만, 그때마다 클러스터들이 서로 합해져 하나의 큰 클러스터를 만들어 제대로 군집화가 안되는 것을 확인할 수 있었다. DBSCAN은 파라미터의 설정에 따라 결과가 크게 달라질 수 있기 때문에, 적절한 파라미터 조합을 찾는 것이 어렵고, 클러스터 개수를 사전에 지정하지 않고 동적으로 결정하기 때문에, 결과적으로 해당 데이터셋에 적절한 클러스터링 결과를 얻는 것이 어렵다고 할 수 있다.

다음으로, K-mean의 경우, 초기 중심점을 무작위로 선택하기 때문에, 초기화에 따라 클러스터링 결과가 달라질 수 있다. 이를 보완하기 위해 여러 번 반복하여, 가장 많은 결과가 나오는 군집화를 사용하기도 하지만, 초기화에 따른 영향을 완전히 제거하기는 어렵다. 또한, 비슷한 개발 수준을 가진 국가들끼리 군집화를 하여, 도움의 우선순위를 정해야 하는데, 이 순위가 매번 달라진다면 객관적이지 못하고, 많은 나라들의 반발이 있을 가능성이 존재한다.

반면, 계층적 군집화는 데이터의 군집화 결과를 덴드로그램을 통해 시각화하는 것이 용이하고, 거리나 유사도를 기반으로 클러스터를 형성하기 때문에 클러스터링 되는 기준이 명확하고 명료하다. 따라서, K-mean의 객관적이지 못하다는 점을 계층적 군집화를 통해 해결할 수 있다. 또한, 클러스터 수를 사전에 정하지 않고, 계층의 높이를 기준으로 클러스터링 결과를 선택할 수 있기 때문에, 상황에 따라 국가들을 소수의 군집 수로 군집화할지, 다수의 군집 수로 군집화할지 변화를 주는 것이 가능하다. 마지막으로, 여러 가지 linkage에 대해 군집화를 할 수 있기 때문에, 다양한 분석적 요구에 맞춰 다양한 수준에서 탐색할 수 있는 기회를 제공한다.

결과적으로, 계층적 군집화가 객관적이면서 상황이나 요구에 맞게 여러 군집화 결과를 낼 수 있다는 점에서, 본 과제의 데이터셋에 가장 적합한 군집화 알고리즘이라고 판단할 수 있다.