Multivariate Data Analysis Assignment #5

Association rule & Clustering

산업경영공학부 2020170831 민찬홍

[Q1] 데이터 변환

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

```
import pandas as pd
data=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/big_student_clear_third_version.csv')
# 필요한 열 선택 및 결합
data['Item Name'] = data[['institute', 'course_id', 'final_cc_cname_DI', 'LoE_DI']].apply(lambda x:' '.join(x.astype(str)),axis=1)

transaction_data=data[['userid_DI', 'Item Name']].copy()
transaction_data.head()
```

Pandas 모듈을 이용하여 csv 파일을 읽어드렸다. 이후 필요한 열을 선택하고 결합하여 연관규칙 분석용 데이터셋인 transcation_data를 만들었다. transcation_data에 대한 상위 5개 행의 값을 출력한 결과는 아래와 같다.

	userid_DI	Item Name
0	MHxPC130313697	HarvardX PH207x India Bachelor's
1	MHxPC130237753	HarvardX PH207x United States Secondary
2	MHxPC130202970	HarvardX CS50x United States Bachelor's
3	MHxPC130223941	HarvardX CS50x Other Middle East/Central Asia ...
4	MHxPC130317399	HarvardX PH207x Australia Master's

[Q2] 데이터 불러오기 및 기초 통계량 확인

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

```
Data shape: (416921, 2)
Data types:
 userid_DI    object
Item Name    object
dtype: object
Missing values:
 userid_DI    0
Item Name    0
dtype: int64
Unique Transaction IDs: 335650
Unique Items: 1405
Average number of items per transaction: 1.232352152539848
            userid_DI                                Item Name
count           416921                                   416921
unique          335650                                     1405
top     MHxPC130386513  MITx_6.00x_United States_Bachelor's
freq                15                                    14412
```

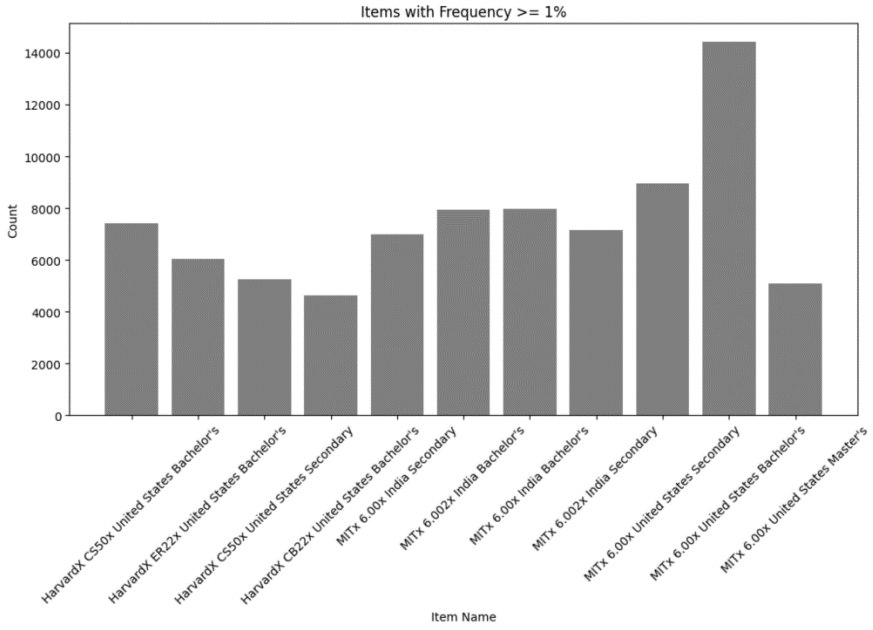
앞서 [Q1]에서 생성한 transcation_data는 2개의 열을 결합하였기에 행의 개수는 본 데이터셋과 동일한 416921개가 나왔으나 열은 2개인 것을 확인할 수 있다. 각각의 열에 대한 데이터 타입을 분석한 결과 두 열 모두 문자열 데이터를 포함하는 것을 확인할 수 있다. 2개의 열에 대한 데이터의 결측치는 존재하지 않았다. 각각의 열에 대한 고유값의 개수는 335650개, 1405개인 것을 확인할 수 있었다. Transaction id당 평균 item의 개수는 약 1.232개 나왔다.

Userid_di행에서 가장 많이 나온 값은 MHxPC130386513였고, Item Name에서 가장 많이 나온 값은 MITx 6.00x United States Bachelor's인 것을 확인할 수 있었다. 각각이 나온 빈도는 15, 14412인 것을 확인할 수 있었다.

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

아이템 빈도 정보를 이용하여 WordCloud를 그린 결과는 위 그림과 같다. 워드클라우드에서 글자의 크기는 해당 단어의 빈도수를 나타내고, 이 때 빈도가 많을수록 글자의 크기가 커진다. 위 그림에서 MITx 6.00x United States Bachelor's가 글자가 제일 커 빈도가 제일 높은 것을 확인할 수 있고 앞서 [Q2-1]의 분석결과와 일치한다는 사실을 워드클라우드 시각화를 통해 다시 한번 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 Chart는 위와 같다. 가로축은 최소 빈도 1% 이상 등장한 Item명, 세로축은 각 Item이 등장한 절대 빈도를 의미한다. 이 때, Item 이름의 길이가 너무 길기 때문에 45도 회전시켰다.

```
Top 5 items and their access countries:
Item: MITx 6.00x United States Bachelor's, Countries: ['United States']
Item: MITx 6.00x United States Secondary, Countries: ['United States']
Item: MITx 6.00x India Bachelor's, Countries: ['India']
Item: MITx 6.002x India Bachelor's, Countries: ['India']
Item: HarvardX CS50x United States Bachelor's, Countries: ['United
States']
```

상위 5개 Item에 대한 접속 국가는 위 그림과 같다. Item 빈도 기준 내림 차순한 결과로 빈도가 높은 순서대로 국가는 United States, United States, India, India, United States인 것을 확인할 수 있었다.

[Q3] 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.

Support는 빈발 아이템 집합을 판별하는데 사용하는 지표로써 지지도가 높을수록 해당 규칙을 적용할 기회가 많아지는 것을 의미한다. 경우에 따라서 P(A)와 P(A,B) 나뉘어 사용되고, 본 문항에서는 apriori 알고리즘을 사용하기에 P(A,B)의 의미로써 Support를 사용하였다. Confidence의 경우 조건부 확률 P(A|B)로써 아이템 집합 간의 연관성 강도를 측정하는데 사용하는 지표이다. 그러나 결과절에 해당하는 아이템이 기본 아이템인 경우 신뢰도가 100%이기에 신뢰도가 높다고 좋은 규칙은 아니다. 좋은 규칙이라고 판단하기 위해서는 지지도, 신뢰도, 향상도 등의 지표를 종합적으로 고려해야 한다.

각 조합 당 최소 10개 이상의 규칙이 생성되도록 support_values의 값을 0.001, 0.002, 0.025 confidence_values의 값을 0.01, 0.05, 0.1로 변화시키며 규칙 개수를 count하였다. 그러나 본 데이터셋의 크기가 큰 관계로 코랩에서 support, confidence에 따른 규칙을 생성 시 계속하여 런타임이 끊기는 상황이 발생하였다. 코드의 문제는 전혀 아니라고 판단하였기에 [Q1]에서 생성한 데이터셋의 10%를 랜덤 샘플링하여 사용하였다. 연관규칙생성을 생성을 위해 샘플링 전 사용한 코드는 아래와 같다(샘플링 전 후 코드는 샘플링 유무 제외 동일하다).

```
# 필요한 열 선택 및 결합
data['Item Name'] = data[['institute', 'course_id',
'final_cc_cname_DI', 'LoE_DI']].apply(lambda x:'
'.join(x.astype(str)),axis=1)

transaction_data=data[['userid_DI', 'Item Name']].copy()

# 트랜잭션 데이터 생성
transactions = transaction_data.groupby('userid_DI')['Item
Name'].apply(list).values.tolist()
# 트랜잭션 데이터 인코딩
te = TransactionEncoder()
te_ary = te.fit_transform(sampled_transactions)
df = pd.DataFrame(te_ary, columns=te.columns_)
```

아래는 트랜잭션 데이터 인코딩을 수행한 뒤 support, confidence 조합에 대해 연관규칙생성을 시도하였으나 실행이 도중에 중단되는 사진이다.

```python
# Support와 Confidence 값의 조합
support_values = [0.001, 0.002, 0.003]
confidence_values = [0.01, 0.05, 0.1]

# 결과 저장을 위한 리스트
results = []

# 규칙 생성 및 결과 확인
for support in support_values:
    for confidence in confidence_values:
        frequent_itemsets = apriori(df, min_support=support, use_colnames=True, verbose=True)
        rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=confidence)
        results.append((support, confidence, len(rules)))
```

```
Processing 62750 combinations | Sampling itemset size 2
```

특별히 일부 데이터만 샘플링하여 연관규칙 생성을 한 가장 큰 이유는 전체 데이터 사용으로 인한 계산 비용 증가와 메모리 문제이다. 특별히 apriori 알고리즘의 경우 후보 아이템셋을 생성하고, 빈도수 기준으로 필터링하는 과정에서 가능한 모든 아이템셋을 탐색해야하는데 이 때 아이템 수가 증가할수록 탐색해야 할 조합의 수는 기하급수적으로 증가한다. 따라서 샘플링을 통해 이 문제점을 보완하고자 하였다.

따라서 sklearn의 train_test_split을 통한 전체 데이터셋의 10% 랜덤 샘플링을 진행하였다. 전체 데이터의 개수인 416921개의 10%인 41692개도 전체 데이터셋의 특성을 잘 반영하면서 연관 규칙을 생성하는데 충분한 크기의 데이터셋이라고 판단하였다. 특별히 랜덤 샘플링을 통해 각 항목이 동일하게 선택되게 함으로써 전체 데이터셋의 대표성을 효과적으로 유지할 수 있을 것이라 생각하였다. 이를 바탕으로 설정한 support, confidence 조합에 대해 생성된 규칙의 수는 아래 표와 같다.

# of Rules	0.01	0.05	0.1
0.001	66	60	41
0.002	20	20	16
0.0025	14	14	12

위 표에서 알 수 있듯이 지지도, 신뢰도 값이 증가함에 따라 생성되는 연관규칙의 개수가 감소하는 것을 알 수 있다.

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

✓ Support가 가장 높은 규칙은 무엇인가?

```
Support가 가장 높은 규칙:
antecedents          (HarvardX CS50x United States Bachelor's)
consequents             (MITx 6.00x United States Bachelor's)
antecedent support                                    0.021868
consequent support                                    0.043021
support                                               0.003724
confidence                                              0.1703
lift                                                  3.958525
leverage                                              0.002783
conviction                                            1.153403
zhangs_metric                                          0.76409
Name: 20, dtype: object
```

Support가 가장 높은 규칙의 조건절과 결과절은 각각 HarvardX CS50x United States Bachelor's, MITx 6.00x United States Bachelor's였다. 이를 해석하면 HarvardX에서 CS50x 강좌를 수강하고 학위 과정이 Bachelor's이며, 접속 국가가 미국인 학생들이 동시에 MITx의 6.00x 강좌를 수강하는 경우가 많다는 것을 의미한다. 각각의 조건절과 결과절이 발생할 확률은 위 사진에서 알 수 있듯이 0.021868, 0.043021이다. 이는 전체 트랜잭션 중 약 2.19%, 4.30%가 각각 조건절과 결과절을 포함하고 있음을 의미한다. support는 규칙 전체의 발생 빈도 비율로 조건절과 결과절이 동시에 발생하는 확률을 의미한다. 본 규칙의 support는 0.003724로 가장 높은 support 값을 가지는 규칙이었다.

나머지 지표에 대해서도 설명하도록 하겠다. Confidence는 조건부 확률로 조건절이 발생할 때 결과절이 발생할 확률을 의미한다. Lift는 조건절과 결과절 사이의 독립성 비율을 나타내는 지표로 1을 기준으로 1보다 크면 두 아이템 집합 사이에 긍정적인 연관관계가 있다고 해석된다. 반대로 1보다 작은 경우는 대체재와 같이 부정적인 연관관계인 경우이고, 1인 경우는 두 아이템 집합이 독립인 것을 의미한다. Leverage는 두 아이템셋이 함께 발생하는 빈도가 두 아이템셋이 독립적일 때 발생하는 빈도와의 차이를 나타내는 값으로 클수록 두 아이템셋이 유의미한 상관 관계가 있는 것이다. Conviction의 경우 조건절이 발생할 때 결과절이 발생하지 않을 확률의 반대되는 값으로 1보다 크면 긍정적 연관성을 의미한다. Zhang's Metric은 연관 규칙의 강도를 나타내는 지표로 높은 값일수록 강한 연관성을 나타낸다고 할 수 있다.

✓ Confidence가 가장 높은 규칙은 무엇인가?

```
Confidence가 가장 높은 규칙:
antecedents           (MITx 8.02x India Bachelor's)
consequents          (MITx 6.002x India Bachelor's)
antecedent support                        0.006405
consequent support                        0.023238
support                                   0.002652
confidence                                0.413953
lift                                     17.813268
leverage                                  0.002503
conviction                                1.666696
zhangs_metric                             0.949947
Name: 44, dtype: object
```

Confidence가 가장 높은 규칙의 조건절과 결과절은 각각 MITx 8.02x India Bachelor's, MITx 6.002x India Bachelor's였다. 이는 MITx에서 8.02x 강좌를 수강하고 학위 과정이 Bachelor's이며, 접속 국가가 인도인 학생들일 때 MITx의 6.002x 강좌를 수강하는 경우가 많다는 것을 의미한다. 나머지 지표들에 대해서는 앞서 설명하였으므로 설명을 생략하도록 하겠다.

✓ Lift가 가장 높은 규칙은 무엇인가?

```
Lift가 가장 높은 규칙:
antecedents          (HarvardX CB22x United States Secondary)
consequents          (HarvardX ER22x United States Secondary)
antecedent support                                   0.007538
consequent support                                   0.009981
support                                              0.001609
confidence                                           0.213439
lift                                                21.385287
leverage                                             0.001534
conviction                                           1.258668
zhangs_metric                                        0.960479
Name: 11, dtype: object
```

Lift가 가장 높은 규칙의 조건절과 결과절은 HarvardX CB22x United States Secondary, HarvardX ER22x United States Secondary였다. 이 때의 support값은 21.385287로 1과 비교하였을 때 매우 큰 값이기에 HarvardX CB22x United States Secondary, HarvardX ER22x United States Secondary가 강한 연관관계를 가지고 있다고 해석할 수 있다.

✓ 만일 하나의 규칙에 대한 효용성 지표를 Support × Confidence × Lift로 정의한다면 효용성이 가 장 높은 규칙 1위~3위는 어떤 것들인가?

```
효용성이 가장 높은 규칙 1위~3위:
                      antecedents                      consequents  ₩
48   (MITx 8.02x India Secondary)   (MITx 6.002x India Secondary)
44  (MITx 8.02x India Bachelor's)  (MITx 6.002x India Bachelor's)
49  (MITx 6.002x India Secondary)    (MITx 8.02x India Secondary)

    antecedent support  consequent support   support  confidence       lift  ₩
48            0.007150            0.019485  0.002920    0.408333  20.956741
44            0.006405            0.023238  0.002652    0.413953  17.813268
49            0.019485            0.007150  0.002920    0.149847  20.956741

    leverage  conviction  zhangs_metric   utility
48  0.002780    1.657209       0.959141  0.024985
44  0.002503    1.666696       0.949947  0.019552
49  0.002780    1.167848       0.971206  0.009169
```

효용성 지표 Support × Confidence × Lift를 Utility라고 정의하면 효용성이 가장 높은 규칙 1~3위는 위와 같다. 1위의 조건절과 결과절은 MITx 8.02x India Secondary과 MITx 6.002x India Secondary , 2위의 조건절과 결과절은 MITx 8.02x India Bachelor's과 MITx 6.002x India Bachelor's, 3위의 조건절과 결과절은 MITx 6.002x India Secondary, MITx 8.02x India Secondary였다. 효용성이 높은 각각의 규칙들에 대한 다양한 효용성 지표들의 값은 위 사진에서 확인할 수 있다.

특별히 규칙의 효용성 정도를 평가할 때 지지도, 신뢰도, 향상도 중 하나를 기준으로 평가하는 것이 아닌 종합적으로 평가해야 한다. 왜냐하면 두 규칙 중 한 규칙이 다른 규칙보다 신뢰도나 지지도나 향상도만 높다고 효용성이 높다고 결론 내릴 수 없기 때문이다.

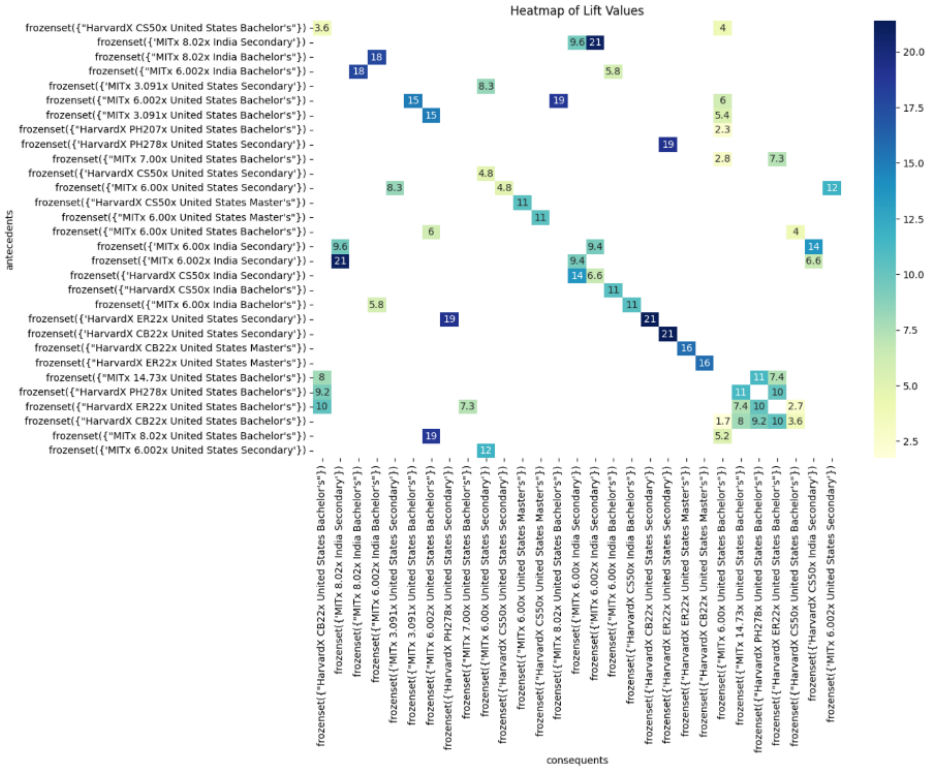
[Extra Question] 이 외 수업 및 실습 시간에 다루지 않은 연관규칙분석 시각화 및 해석을 시도해 보시오.

앞서 [Q3-2]에서 생성된 효용성이 가장 높은 연관규칙인 MITx 8.02x India Secondary -> MITx 6.002x India Secondary에 대해서 수업 시간에 다루지 않은 효용성 지표들로 분석을 하겠다. Leverage는 두 아이템셋이 함께 발생하는 빈도가 두 아이템셋이 독립적일 때 발생하는 빈도와의 차이를 나타내는 값으로 클수록 두 아이템셋이 유의미한 상관 관계가 있는 것을 의미한다. 이 때의 Leverage는 0.002780으로 두 아이템셋이 어느 정도 함께 발생하는 빈도가 더 높다고 해석할 수 있다. Conviction의 경우 조건절이 발생할 때 결과절이 발생하지 않을 확률의 반대되는 값으로

1보다 크면 긍정적 연관성을 의미한다. Conviction 값이 1.657209로 1보다 크기에 두 아이템셋이 긍정적 연관성을 가지고 있다고 할 수 있다. Zhang's Metric은 연관 규칙의 강도를 나타내는 지표로 높은 값일수록 강한 연관성을 나타낸다고 할 수 있다. 이 때의 Zhang's Metric은 0.959141로 두 아이템 셋이 강한 연관성을 가지고 있다고 할 수 있다.

효용성 지표 Support × Confidence × Lift를 Utility로 정의했을 때 Utility가 높은 규칙들은 수업 시간에 다르지 않은 Leverage, Conviction, Zhang's Metric 관점에서 봤을 때도 강한 연관관계가 있는 것을 확인할 수 있었다. 그러나 Utility가 높은 순으로 다른 효용성 지표들의 값(Leverage, Conviction, Zhang's Metric) 역시 높은 것은 아닌 것을 확인할 수 있었다. 이는 Utility가 Support, Confidence, Lift를 종합적으로 고려한 지표이기 때문에 나타난 결과라고 해석된다.

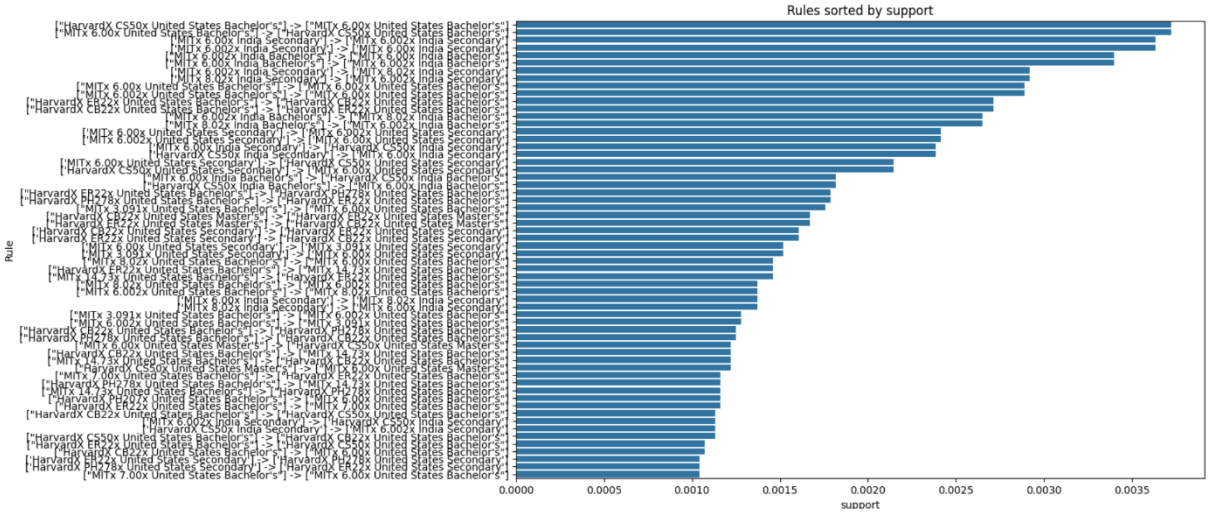
수업 시간에 다루지 않은 연관규칙분석 시각화는 앞서 [Q3-2]에서 진행한 지지도 0.001, 신뢰도 0.05인 규칙들에 대해 시각화를 진행하도록 하겠다.

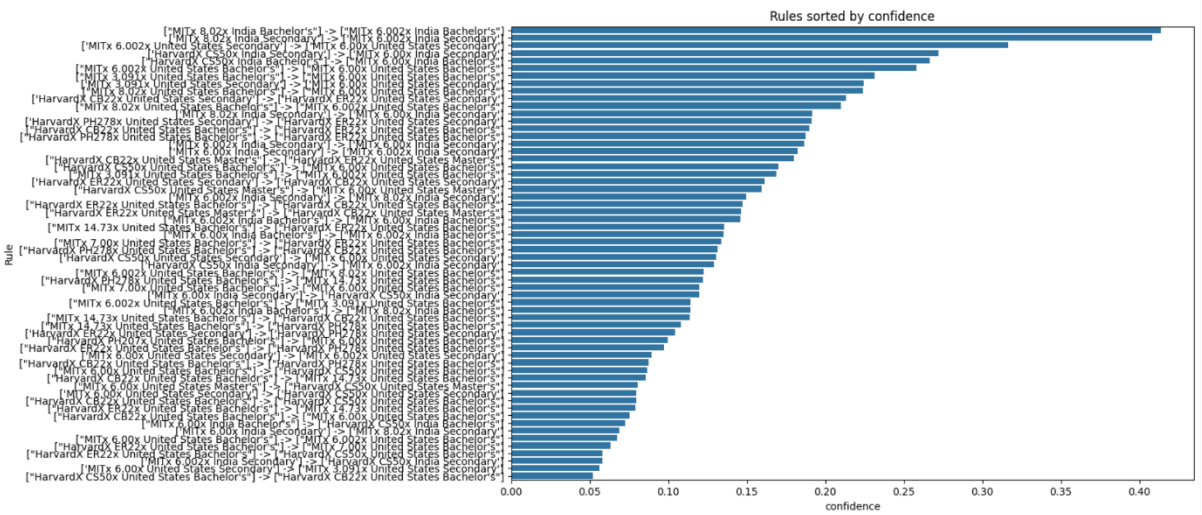


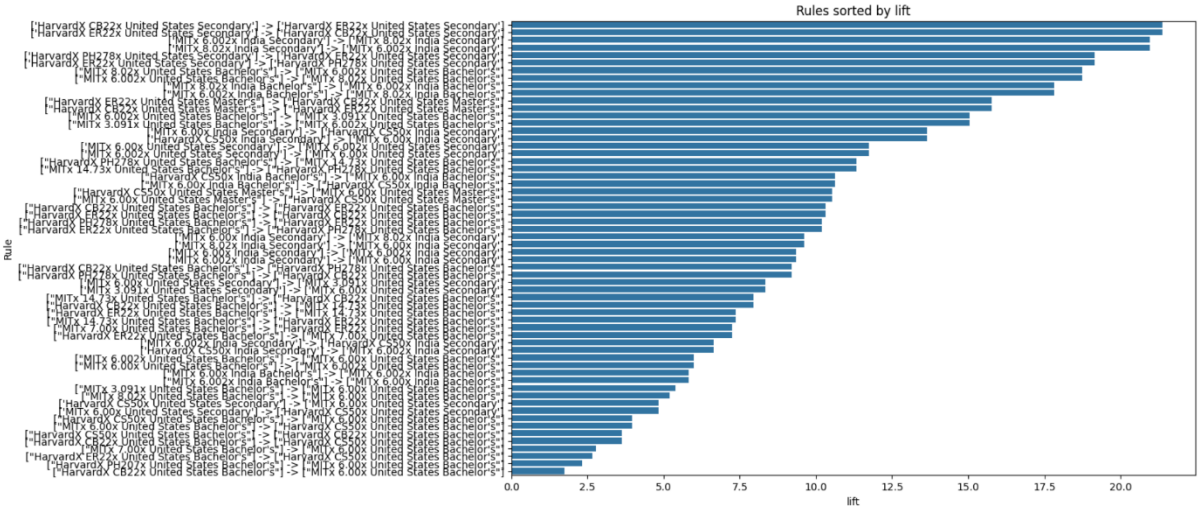
먼저 히트맵으로 시각화를 진행해보았다. 히트맵(Heatmap)은 행과 열의 값에 따라 색상으로 데이터를 시각화하는 도구이다. 연관 규칙 분석에서 히트맵은 주로 규칙의 강도를 시각적으로 표현하는 데 사용된다. 위 히트맵에서는 Lift 값을 사용함으로써 이를 통해 연관 규칙 간의 상관관계를 시각적으로 쉽게 파악할 수 있다. 행은 조건절, 열은 결과절을 의미한다. 이 때 색상은 lift 값으로 진할수록 lift 값이 높음을 의미한다.

이외의 시각화 방식으로 네트워크 그래프와 막대 그래프를 사용하여 규칙들을 분석해보고자 하였다. 네트워크 그래프란 노드와 엣지로 구성되어 있는 그래프로 각 노드는 특정한 아이템셋을 노드 간의 연결선은 연관 규칙을 나타내며, 조건절과 결과절 사이의 관계를 보여준다. 엣지의 라벨은 lift와 confidence 값을 나타낸다.

연관 규칙들의 개수가 60개로 다소 많아 눈으로 연관규칙 간 관계를 해석하기에는 다소 어려움이 있는 것으로 보인다. 더 적은 연관 규칙들로 네트워크 그래프 시각화를 수행한다면 시각적인 해석이 좀 더 용이할 것으로 생각된다. 두 노드의 연결은 두 강좌 간 연관성을 의미하고, 이 때 연결된 엣지 위의 lift와 confidence 값을 통해 두 노드의 연관 강도를 알 수 있는 것이다.







또 막대 그래프를 이용하여 연관 규칙을 분석하고자 하였다. 세 그래프는 위에서부터 지지도, 신뢰도, 향상도 순으로 규칙들이 정렬된 그래프를 의미한다. x축은 연관 규칙 지표 값(지지도, 신뢰도, 향상도)를 의미하고, y축은 각 연관 규칙의 조건절과 결과절을 의미한다. 이를 통해 효용성 지표가 높은 연관규칙을 한 눈에 알아볼 수 있다.

[Part 2: Clustering]

[Q1] 데이터셋 선정(ok)

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

데이터셋 링크: https://www.kaggle.com/datasets/harrywang/wine-dataset-for-clustering

본 데이터셋은 'wine-clustering dataset'으로 178 개의 행과 13 개의 열로 구성 되어있다. 각각의 열은 와인의 화학적 성분으로 구성 되어있으며 알코올, 산도, 설탕 함량 등을 포함하고 있어 화학적 성분의 특성의 다차원 분석을 통한 와인의 클러스터링을 하기에 적합한 데이터셋이라고 판단하였다. 왜냐하면 와인의 화학적 성분은 와인의 맛, 향, 품질에 직접적인 영향을 미치기에 클러스터링을 통해 와인의 품질(고품질 와인과 저품질 와인) 또는 종류(레드 와인, 화이트 와인 등)를 분류하는 것이 유의미하다고 생각되었기 때문이다. 특별히 열은 와인의 13 개 화학적 성분, 즉 다양한 특성 변수로 구성 되어있기에 각 와인 종류 간 미묘한 차이를 더 정밀하게 분석하고 이해할 수 있을 것이라 판단하였다.

단순히 효과적인 클러스터링 뿐만 아니라 이를 와인 산업에 적용할 수 있을 것이라 생각하였다. 각 군집에 따른 알코올, 산도, 설탕 함량 등의 성분이 다르기에 소비자는 자신이 선호하는 와인 종류를 더 쉽게 찾을 수 있을 것이라 기대된다. 단순히 소비자 뿐만 아니라 와인 생산 측면에서도 기대 효과를 가질 것이라 생각하였다. 특정 클러스터가 시장에서 큰 인기를 끌고 있다면, 그 클러스터에 속한 와인의 생산에 더 많은 자원을 투입하여 수익을 극대화할 수 있을 것이다. 즉, 클러스터링 결과를 바탕으로 자원을 효율적으로 배분할 수 있다. 또한 와인의 품질을 일정하게 유지하거나 향상시키기 위해 클러스터링 결과를 활용할 수 있다. 특정 클러스터가 높은 품질을 나타낸다면, 해당 클러스터에 속한 와인의 생산 과정이나 원재료를 분석하여 품질 향상을 위한 최적의 방법을 찾을 수 있을 것이라 생각하였다.

[Q2] 최적의 군집 수 선정

[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 얼 마인가? Silhouette index 기준으로 가장 최적의 군집 수는 몇 개로 판별이 되었는가?

최적의 군집 수를 찾기에 앞서 데이터의 scaling 작업을 수행하였다. 본 데이터셋은 다양한 특성으로 이루어져 있기에 특성의 크기가 다를 수밖에 없다. 이 때 크기가 큰 특성의 경우 클러스터링 과정에서 더 큰 영향을 미치기에 이를 방지하고자 scaling 작업을 수행하였다. 구체적으로 K-means 알고리즘은 클러스터 중심(centroid)까지의 유클리드 거리를 최소화하는 방식으로 동작하는데 이 때 거리 계산 시 모든 특성이 동일 중요도로 고려되도록 하였다. 이는 알고리즘이 안정적이고 빠른 수렴으로 이어진다.

군집화 타당성 지표로 1)Dunn Index와 2) Silhouette index 값을 산출하였다. Dunn Index란 군집 간 거리 중 최소값을 군집의 지름 중 가장 큰 값으로 나눈 지표이다. 이 때, 군집화가 잘 되면 잘 될수록 군집 간 거리는 멀 것이고, 군집의 지름은 작을 것이기에 Dunn Index값이 클수록 군집화가 잘 되었다고 판단할 수 있다. Silhouette index란 (개체 i로부터 다른 군집내에 있는 개체들 사이의 평균 거리 중 최소 값(b(i))-개체 i로부터 같은 군집 내에 있는 모든 다른 개체들 사이의 평균 거리(a(i)))를 a(i), b(i)의 최대값으로 나눈 지표를 의미한다. -1이상 1이하의 값을 가지며, 이 지표 역시 Dunn Index와 동일한 논리로 값이 클수록 군집화가 잘 되었다고 판단할 수 있다. 특별히 군집화의 경우 비지도 학습으로 완벽한 정답이 없기에 다양한 군집화 타당성 지표를 고려해야 한다고 판단하였고, Silhouette index뿐만 아니라 Dunn Index 값도 산출하였다. Dunn Index의 경우 python에서 별도의 라이브러리가 존재하지 않기에 아래와 같이 Dunn Index를 산출하기 위한 함수를 정의하였다.

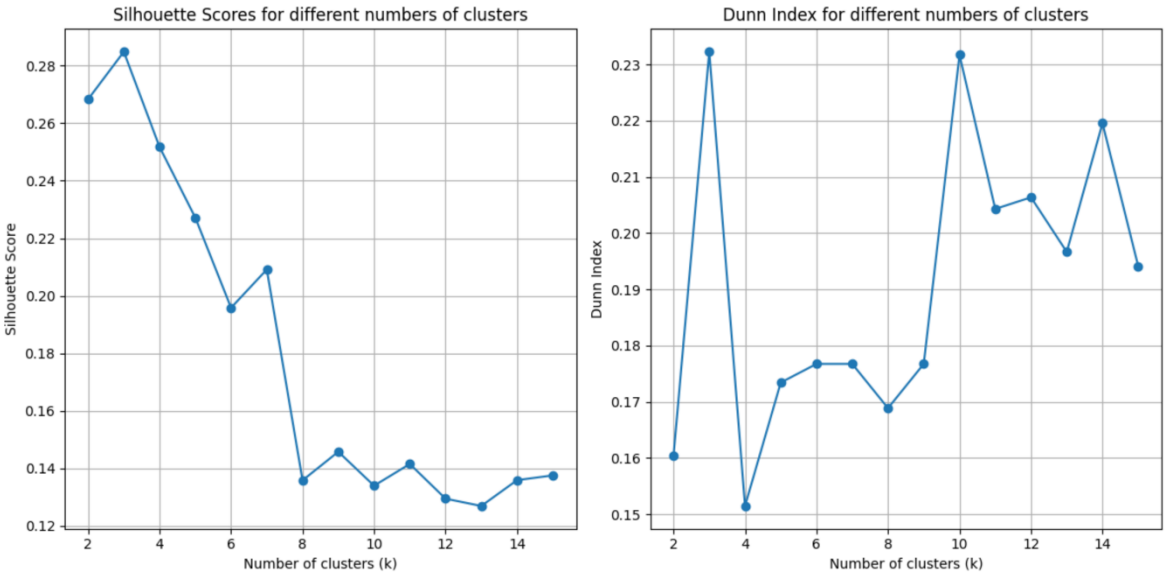
```
# Dunn Index 계산 함수
def dunn_index(data, labels):
    """
    Dunn Index를 계산하는 함수.
    :param data: 입력 데이터 (numpy array)
    :param labels: 각 데이터 포인트의 군집 레이블 (numpy array)
    :return: Dunn Index 값 (float)
    """
    unique_cluster_labels = np.unique(labels)
    num_clusters = len(unique_cluster_labels)

    if num_clusters == 1:
        return 0

    # 클러스터 간 최소 거리 계산
    inter_cluster_distances = []
    for i in range(num_clusters):
        for j in range(i + 1, num_clusters):
            cluster_i = data[labels == unique_cluster_labels[i]]
            cluster_j = data[labels == unique_cluster_labels[j]]
            inter_cluster_distances.append(np.min(pairwise_distances(clu
ster_i, cluster_j)))

    # 클러스터 내 최대 거리 계산
    intra_cluster_distances = []
    for i in range(num_clusters):
        cluster_i = data[labels == unique_cluster_labels[i]]
        intra_cluster_distances.append(np.max(pairwise_distances(cluster
_i, cluster_i)))

    return np.min(inter_cluster_distances) /
np.max(intra_cluster_distances)
```

최적의 군집 수 탐색을 위해 군집 수는 2부터 15까지 증가 폭을 1씩 증가하며 탐색을 진행하였다. 먼저 군집 수 탐색을 15까지 진행한 이유는 너무 많지도 적지도 않은 군집 탐색 수라고 판단하였기 때문이다. 너무 많은 군집 수를 탐색하면 계산 비용이 증가하고, 너무 적은 군집 수를 탐색하면 최적의 군집 수를 찾지 못할 가능성이 존재한다. 이 때, 본 데이터셋이 178개의 행으로 구성 되어있고, 군집 수 탐색 시 통상적으로 데이터 개수의 약 10%까지 탐색을 진행하기에 15까지 탐색을 진행한 것이다. 2~15까지 군집 수 탐색 범위가 많지 않다고 판단하였고, 본 데이터셋이 13개의 다양한 특성을 가지고 있기에 보다 정밀한 최적 군집 수 탐색을 위해 탐색할 군집 수 증가폭을 1로 설정하였다. 군집 수에 따른 1)Dunn Index와 2) Silhouette index는 아래와 같다.

군집 수에 따른 Dunn Index와 Silhouette Score

K	Dunn Index	Silhouette Score
2	0.16040	0.26831
3	0.23226	0.28486
4	0.15149	0.25173
5	0.17344	0.22717
6	0.17675	0.19582
7	0.17675	0.20913
8	0.16889	0.13582
9	0.17675	0.14576
10	0.23169	0.13395
11	0.20431	0.14151
12	0.20637	0.12944
13	0.19667	0.12690
14	0.21955	0.13587
15	0.19412	0.13754

이 때, 실루엣 지표가 가장 크게 나온 경우가 최적의 군집 수가 되므로 실루엣 지표 기준 최적의 군집 수는 3인 것을 확인할 수 있었다. 최적의 군집을 탐색하는 데 걸린 시간 2.76771초였다.



위는 클러스터 수에 따른 Silhouette index와 Dunn Index의 변화 양상을 나타낸 그래프이다. 그래프를 보면 알 수 있듯이 전반적으로 Silhouette index는 감소 추세를 띈다. 통상적으로 b(i)>a(i)이기에 이 경우라고 가정할 시 Silhouette index는 감소 추세는 군집 수가 증가함에 따라 다른 군집 내 개체들 사이의 평균 거리가 감소하고, 같은 군집 내 개체들 사이의 평균 거리가 증가하였기 때문이라고 해석된다. 또 다른 말로 a(i)/b(i)값이 증가하여 1-a(i)/b(i)값이 감소한 것이다.

Dunn Index 역시 Silhouette index와 마찬가지로 클러스터의 수가 3일 때 최대값을 가지는 것을 확인할 수 있다. 즉, 또 다른 군집 타당성 지표의 결과도 동일한 것을 알 수 있다. 이는 최적의 군집 수가 3이라는 사실을 더 강하게 뒷받침해주는 근거라고 생각한다.

[Q3] 군집화 반복

[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 몇 번 반복되어 발생하는지 확인해보시오.

[Q2]에서 선택된 최적의 군집의 수는 3이었다. 이를 바탕으로 K-Means Clustering을 10회 반복한 결과 아래와 같이 2개의 군집화 결과가 나왔다. 각 결과물에 대한 Size 역시 보이기 위해 마지막 행에 Size 행을 추가하였다. 각 군집화 결과물의 centroid 및 size, 반복 횟수는 아래 표를 통해 확인할 수 있다.

결과물 1:

	Cluster 1	Cluster 2	Cluster 3
Alcohol	0.164907	-0.939003	0.878097
Malic_Acid	0.871547	-0.391966	-0.304576
Ash	0.186898	-0.439201	0.318942
Ash_Alcanity	0.524367	0.208988	-0.664524
Magnesium	-0.075473	-0.463774	0.564888
Total_Phenols	-0.979330	-0.053348	0.876505
Flavanoids	-1.215248	0.066904	0.943639
Nonflavanoid_Phenols	0.726064	-0.019822	-0.585590
Proanthocyanins	-0.779706	0.064792	0.581783
Color_Intensity	0.941539	-0.882075	0.167188
Hue	-1.164789	0.452982	0.483728
OD280	-1.292412	0.289738	0.767053
Proline	-0.407088	-0.756026	1.158347
Size	51.000000	66.000000	61.000000

이 군집화 결과물은 총 1 번 반복되었습니다.

결과물 2:

	Cluster 1	Cluster 2	Cluster 3
Alcohol	-0.926072	0.164907	0.835232
Malic_Acid	-0.394042	0.871547	-0.303810
Ash	-0.494517	0.186898	0.364706
Ash_Alcanity	0.170602	0.524367	-0.610191
Magnesium	-0.491712	-0.075473	0.577587
Total_Phenols	-0.075983	-0.979330	0.885237
Flavanoids	0.020813	-1.215248	0.977820
Nonflavanoid_Phenols	-0.033534	0.726064	-0.562090
Proanthocyanins	0.058266	-0.779706	0.580287
Color_Intensity	-0.901914	0.941539	0.171063
Hue	0.461804	-1.164789	0.473984
OD280	0.270764	-1.292412	0.779247
Proline	-0.753846	-0.407088	1.125185
Size	65.000000	51.000000	62.000000

이 군집화 결과물은 총 9 번 반복되었습니다.

K-Means Clustering 을 10 회 반복하여 나온 두 가지 군집화 결과 중 하나의 결과가 9 번 반복되었다. 이는 9 번 반복된 결과가 매우 안정적이고 데이터의 구조를 잘 반영하는 데이터의 분포를 가장 잘 설명하는 군집화 결과라고 해석할 수 있다. 동시에 데이터가 명확하게 구분되는 군집을 가지고 있어 군집 경계가 분명하다고 해석된다. 그럼에도 1 번의 다른 군집화 결과물이 나온 것은 이상치나 노이즈에 의해 발생했을 것이라 생각된다. 무엇보다 반복된 결과가 매우 일관되게 나타나기에 Silhouette index 기준 선택한 최적의 군집 수(K=3)를 잘 설정했다고 할 수 있고, 해당 군집화 결과를 신뢰할 수 있다는 것을 의미한다.

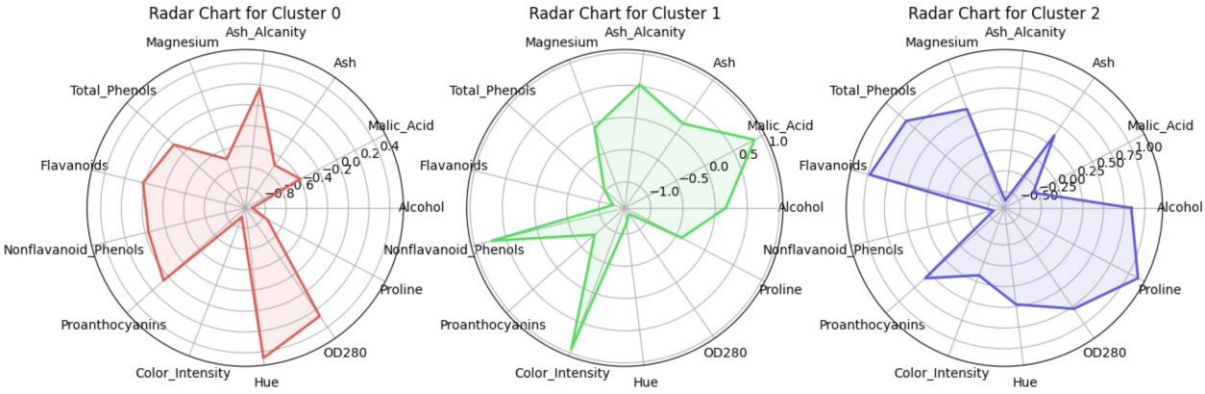
군집화 결과를 산출하는 과정에서 두 군집화 결과의 중심점이 동일하나 cluster 순서가 뒤바뀐 경우 다른 군집화 결과로 취급하는 상황을 방지하기 위해 아래 클러스터 중심점 비교 함수를 사용하였다.

```
# 클러스터 중심점 비교 함수
def are_clusters_identical(centroids1, centroids2, tolerance=1e-5):
    """
    두 군집화 결과의 중심점이 동일한지 확인하는 함수.
    중심점 간의 유클리드 거리가 tolerance 이하인 경우 동일한 것으로 간주.

    :param centroids1: 첫 번째 군집 중심점 (numpy 배열)
    :param centroids2: 두 번째 군집 중심점 (numpy 배열)
    :param tolerance: 비교를 위한 허용 오차
    :return: 두 군집 중심점이 동일한지 여부 (bool)
```

```
    """
    centroids1_sorted = np.sort(centroids1, axis=0)
    centroids2_sorted = np.sort(centroids2, axis=0)

    return np.allclose(centroids1_sorted, centroids2_sorted,
atol=tolerance)
```

[Q4] Radar Chart

[Q4] [Q3]에서 가장 빈번하게 발생한 군집화 결과물에 대해서 각 군집 별 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.



Radar Chart란 여러 변수의 값을 하나의 차트에 시각화하여 비교할 수 있는 도구이다. 각 축은 변수 하나를 나타내고, 중심에서 바깥으로 뻗어 나가는 각 축은 변수의 값을 나타내며, 축의 끝점을 연결하여 다각형을 형성한다.

Radar Chart 로 판단 시 군집의 속성이 가장 상이할 것으로 예상되는 군집은 군집 0 과 군집 2 이다. 각각의 특성 값을 비교해보면 Alcohol 은 Cluster 0 이 낮고, Cluster 2 가 높다. Malic_Acid 은 Cluster 0 이 높고, Cluster 2 가 낮다. Total_Phenols 와 Flavanoids 는 Cluster 0 이 낮고, Cluster 2 가 높다. 이처럼 여러 특성에서 가장 큰 차이를 보이기에 군집 0 과 군집 2 가 가장 상이할 것이라고 생각하였다.

Radar Chart 로 판단 시 군집의 속성이 가장 유사할 것으로 예상되는 군집은 군집 1 과 군집 2 이다. 대부분의 특성 값이 비슷하며, 구체적으로 Alcohol, Ash, Ash_Alcalinity, Magnesium 의 값이 비슷하다.

[Q5] Paired t-test

[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 선택 된 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?

먼저 군집의 속성이 가장 상이할 것으로 예상되는 군집인 군집 0과 군집 2의 test 결과는 아래와 같다.

```
Results for Cluster 0 vs Cluster 2:
                 Variable     Two_Sided       Greater          Less
0                 Alcohol  1.018635e-14  1.000000e+00  5.093173e-15
1              Malic_Acid  1.475032e-10  1.000000e+00  7.375160e-11
2                     Ash  1.075500e-04  9.999462e-01  5.377501e-05
3           Ash_Alcanity  2.114410e-02  9.894280e-01  1.057205e-02
4              Magnesium  1.118367e-02  9.944082e-01  5.591835e-03
5          Total_Phenols  3.791205e-10  1.895603e-10  1.000000e+00
6             Flavanoids  2.146292e-25  1.073146e-25  1.000000e+00
7   Nonflavanoid_Phenols  7.509477e-05  9.999625e-01  3.754739e-05
8        Proanthocyanins  7.432607e-07  3.716303e-07  9.999996e-01
9        Color_Intensity  4.041358e-18  1.000000e+00  2.020679e-18
10                   Hue  3.690316e-22  1.845158e-22  1.000000e+00
11                 OD280  1.850143e-30  9.250715e-31  1.000000e+00
12               Proline  2.031718e-05  9.999898e-01  1.015859e-05

Proportion of significant differences (p < 0.05): 1.00000
```

모든 변수들에 대한 양측 검정 p-value값이 유의수준 0.05보다 작기에 [Q4]의 가장 상이할 것이라고 예상되는 군집 예측 결과에 부합한다고 알 수 있다. 각각의 변수의 관점에서 위 표를 더 구체적으로 해석하면 Alchohol의 경우 Less의 p-value값이 유의수준 0.05보다 작으므로 군집 0의 Alcohol 도수가 군집 2의 Alcohol 도수보다 낮다고 해석할 수 있다. 앞서 [Q4] Radar chart에서 살펴본 결과와 동일한 것을 다시 한번 확인할 수 있다. 다른 변수들에 대해서도 Greater와 Less 중 p-value 값이 유의수준 0.05보다 작은 부분을 보고 동일한 방식으로 해석할 수 있다.

```
Results for Cluster 1 vs Cluster 2:
                Variable     Two_Sided       Greater          Less
0                Alcohol  5.434328e-07  2.717164e-07  9.999997e-01
1             Malic_Acid  9.208564e-11  1.000000e+00  4.604282e-11
2                    Ash  2.470769e-01  1.235384e-01  8.764616e-01
3          Ash_Alcanity  2.038110e-10  1.000000e+00  1.019055e-10
4              Magnesium  8.488953e-05  4.244477e-05  9.999576e-01
5          Total_Phenols  1.335651e-33  6.678255e-34  1.000000e+00
6            Flavanoids  3.431999e-52  1.715999e-52  1.000000e+00
7   Nonflavanoid_Phenols  4.419198e-12  1.000000e+00  2.209599e-12
8        Proanthocyanins  2.187865e-16  1.093932e-16  1.000000e+00
9        Color_Intensity  6.616898e-06  9.999967e-01  3.308449e-06
10                   Hue  4.995730e-31  2.497865e-31  1.000000e+00
11                 OD280  3.056205e-46  1.528103e-46  1.000000e+00
12               Proline  4.335239e-25  2.167619e-25  1.000000e+00

Proportion of significant differences (p < 0.05): 0.92308
```

위 표는 [Q4]의 가장 유사할 것이라고 예상되는 군집 예측 결과이다. 변수들 중 양측 검정 수행 시 유의수준 0.05보다 큰 변수는 Ash 1개였다. 따라서 유의수준 0.05하에 각 변수 별 평균값 차이가 유의하게 나는 변수는 Ash 제외 11개로 약 92.308%의 변수가 유의하게 차이가 난다고 할 수 있다. 앞서 군집 0과 군집 2 비교 결과와 다르게 가장 유사할 것이라 예상한 군집 결과에 대해 통계적 검정을 시행했기에 차이가 나타나는 변수의 비중이 더 작아진 것을 확인할 수 있다. 군집 0과 군집 2 비교 방식처럼 동일한 방식으로 하나의 변수에 대해 어느 군집의 값이 유의하게 큰지 혹은 작은지 해석할 수 있다.

[Q6] 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?

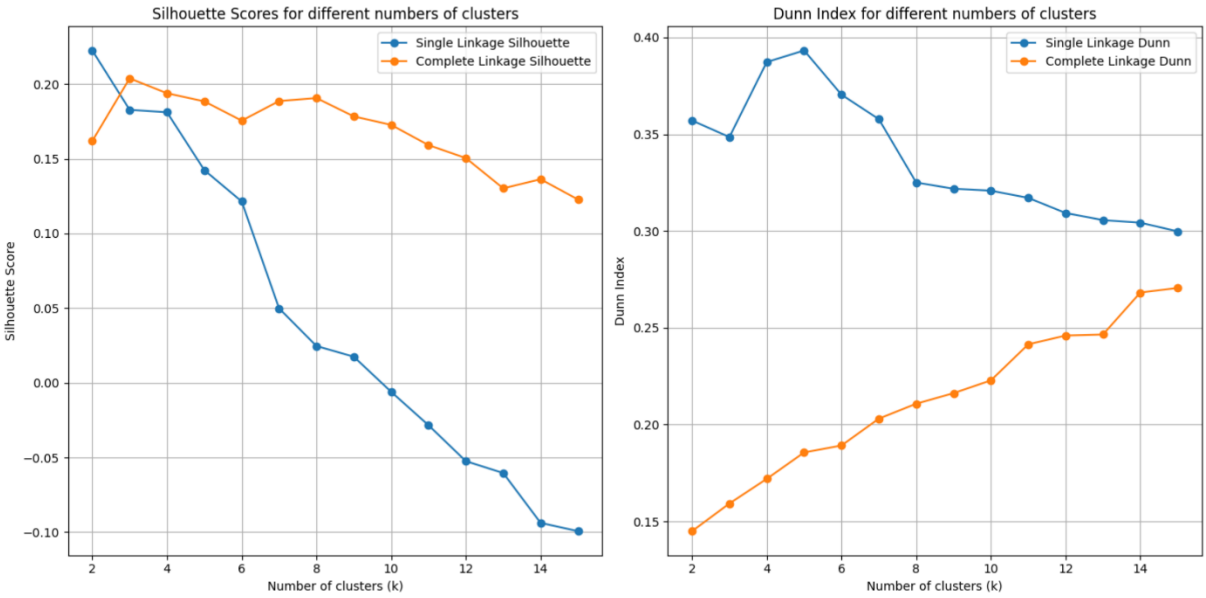
Hierarchical clustering을 하기에 앞서 두 객체 사이의 유사도를 측정하는 지표를 유클리드 거리로 선정하였다. 유클리드 거리란 두 점 사이의 거리로 직관적으로 이해하기 쉽고, 간단하다는 장점이 있다. 또한 본 데이터셋이 연속형 변수로 이루어져 있기에 객체 간 실제 차이를 반영하는데 유리하다고 판단하였다. 그러나 특성마다 클러스터링 과정에서 미치는 영향이 다르고, 차원의 저주라는 유클리드 거리의 단점을 예방하고자 Scaling을 진행하였다.

최적의 군집 수 탐색과 군집 증가 폭은 앞서 본 데이터셋의 특성에 따른 K-Means clustering과 동일한 이유로 2~15까지 1씩 증가시키며 탐색을 진행하였다. 군집 수 변화에 따른 Single Linkage,

Complex Linkage의 군집화 타당성 지표 값은 아래와 같다.

Number of Clusters	Single Linkage Dunn	Complete Linkage Dunn	Number of Clusters	Single Linkage Silhouette	Complete Linkage Silhouette
2	0.357084	0.144968	2	0.222451	0.161868
3	0.348535	0.159244	3	0.182738	0.203787
4	0.387281	0.171983	4	0.181078	0.193825
5	0.393256	0.185626	5	0.142353	0.188365
6	0.370425	0.189196	6	0.121382	0.175491
7	0.357778	0.203074	7	0.049821	0.188596
8	0.325059	0.210784	8	0.024630	0.190601
9	0.321832	0.216243	9	0.017473	0.178327
10	0.320829	0.222837	10	-0.006022	0.172643
11	0.317110	0.241460	11	-0.028394	0.159107
12	0.309302	0.246005	12	-0.052440	0.150425
13	0.305635	0.246535	13	-0.060375	0.130167
14	0.304297	0.268218	14	-0.093734	0.136276
15	0.299809	0.270588	15	-0.099277	0.122725

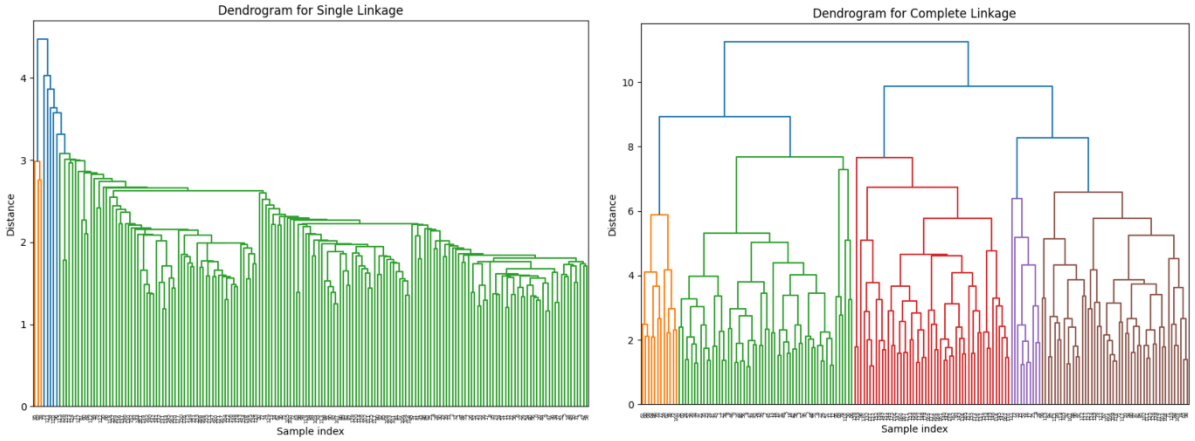
Silhouette index 기준으로 가장 최적의 군집 수는 Simple Linkage에서 2개, Complete Linkage에서 3개이다.



위 그래프는 군집 수에 따른 Silhouette index 와 Dunn Index의 변화 추이를 나타낸 그래프이다. Silhouette index의 경우 두 거리 계산법에 대해 군집 개수가 증가함에 따라 모두 감소하는 추세를 보인다. 이는 군집의 개수가 증가함에 따라 클러스터링 품질이 떨어진다는 신호로 해석할 수 있다. 이는 클러스터가 과도하게 세분화되어 클러스터 간의 분리도가 낮아지고, 클러스터 내의 데이터 포인트 수가 줄어들면서 클러스터 내 분산이 증가하여 발생하는 문제라고 해석된다.

일반적으로 Silhouette Index의 최대값을 가지는 군집 수가 최적의 군집 수로 간주되며, Dunn Index는 이를 보완하는 지표로 사용되고 문제에서 Silhouette index 기준으로 가장 최적의 군집 수를 판별하라 하였기에 Silhouette index 기준으로 최적의 군집 수를 판별하겠다. 이 때, 전반적인 Silhouette index가 complete의 경우가 single보다 높고, 어느 정도 군집의 세분화(적당한 군집 수)가 되어있어야 유의미하게 군집화가 이루어졌다 생각하였기에 최적의 군집 수 산정 시

complete linkage을 기준으로 판별하겠다. 그 결과 최적의 군집 수는 3이라는 결론을 내릴 수 있다. 추가로 본 데이터셋이 다차원 데이터셋이고 잠재적 노이즈를 포함하고 있을 가능성이 있기에 이 상황에서는 이상치나 노이즈에 상대적으로 덜 민감한 complete linkage 계산법이 더 적합하다고 판단하였다. 그래서 실제로 complete linkage가 single linkage의 Silhouette index보다 거의 대부분 값이 높은 것을 확인할 수 있는 것이다.

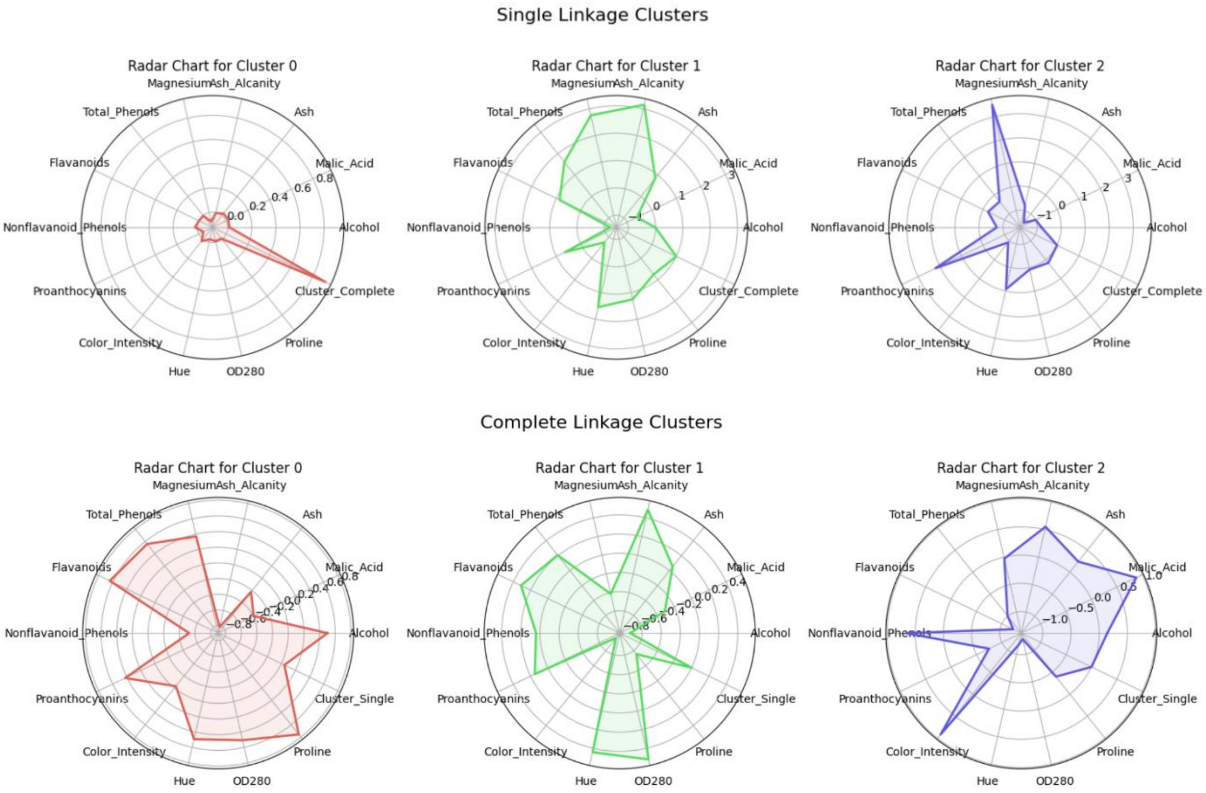


위는 각각 single linkage, complete linkage 방식에 따른 생성된 덴드로그램이다. K Means clustering의 초기 군집 수를 설정하는 것과는 다르게 Hierarchical clustering의 경우 한 번 군집화를 한 뒤 원하는 군집 수에 따라 적절히 덴드로그램을 분할하면 된다. 이 때, 덴드로그램의 높이는 개체 간 거리를 의미하는데 높이가 낮을수록 두 개체가 먼저 합쳐진 것을 의미한다. 즉, 어떤 개체가 어떤 순서로 합쳐졌는지 덴드로그램을 통해 확인할 수 있는 것이다.

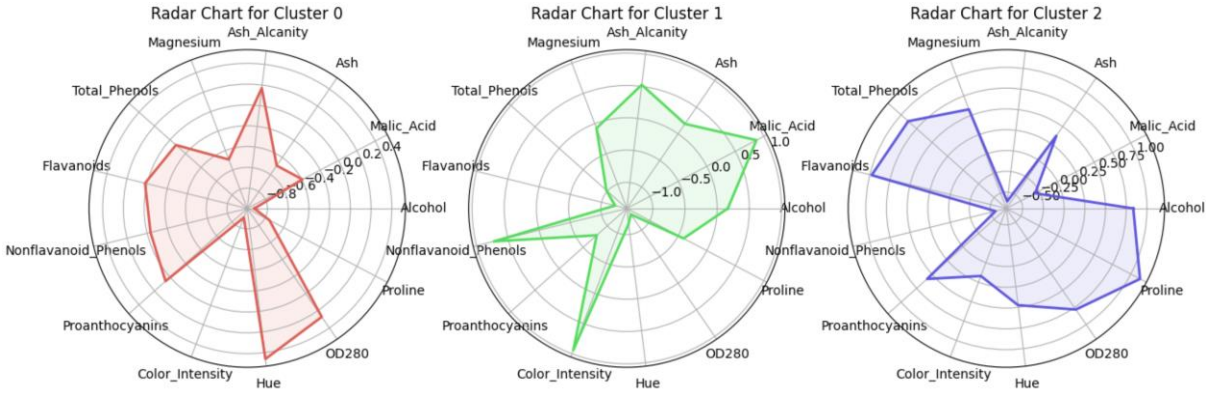
[Q7] Radar chart

[Q7] [Q6]에서 찾은 최적의 군집 수에 대해서 각 군집들의 변수 값의 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.

앞서 [Q6]에서 최적의 군집 수를 3이라고 결론 내렸기에 각각의 거리계산법에 대해 군집 수를 3으로 설정하고 Radar chart를 도사하겠다.



위는 Simple Linkage, Complete Linkage 방식에 따른 계층적 군집화를 묘사한 Radar chart이다.



위는 K-Means Clustering Radar Chart이다. Radar Chart를 바탕으로 판단할 때, K-Means Clustering과 보다 유사한 결과물이 나오는 방식은 Complete Linkage 방식인 것을 알 수 있다. Complete Linkage의 경우 Single Linkage와 다르게 클러스터 병합 시 두 클러스터들 간 가장 먼 점 사이를 최소화하려고 한다. 이는 결과적으로 클러스터 내의 데이터 포인터들이 비교적 가까운 거리에 있게 한다. 즉, 클러스터 내 분산을 최소화하려고 하고 결과적으로 클러스터 중심에서 데이터 포인트들이 멀어지지 않도록 최적화하기에 구형에 가까운 클러스터가 형성된다. 반면, Simple Linkage의 경우 클러스터 병합 시 두 클러스터들 간 가장 가까운 거리를 기준으로 병합하

기에 연쇄적으로 클러스터링이 일어나 chain형태에 가까운 클러스터가 형성된다. 즉, Complete Linkage와 다르게 상대적으로 데이터 포인터들이 넓게 퍼져 있는 클러스터가 생성되는 것이다.

한편 K-Means 클러스터링의 경우 클러스터 내의 데이터 포인트들이 클러스터 중심에 가깝게 모이도록 최적화하여 분산을 최소화하는 방식이다. 즉, 직접적으로 클러스터 내 분산을 최소화하는 방식이기에 구형 클러스터를 형성하려는 경향이 있다. Complete Linkage는 K-Means와 마찬가지로 클러스터 내의 데이터 포인트들이 서로 가깝게 모여 있도록 하여 구형의 클러스터를 형성하려는 경향이 있기에 Complete Linkage는 K-Means 클러스터링과 유사한 결과를 보인 것으로 해석된다.

[Q8] DBSCAN

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

앞서 [Q2]에서 선정한 최적 군집 개수는 3이었다. eps의 경우 0.2~3.0까지 0.2 증가폭을 가지고 탐색을 진행하였고, minPts의 경우 본 데이터셋의 크기가 178개임을 고려하여 앞서 군집화와 동일한 이유로 2부터 15까지 1의 증가폭을 가지고 탐색을 진행하였다. 이 때, eps이란 인접한 데이터를 군집으로 묶기 위한 반지름 파라미터이고, minPts란 eps 내에 minPts개의 데이터가 존재할 때 core point인지를 결정하는 파라미터이다. 선정된 core point는 군집의 중심을 형성하며, 이를 기준으로 군집이 확장된다. DBSCAN의 경우 eps과 minPts를 지정하면 군집 수는 자동적으로 결정되기에 eps, minPts의 값을 조절하며 군집화를 진행하는 과정은 필수적이며 매우 중요하다고 할 수 있다.

앞서 [Q2]에서 선정한 최적 군집 개수 3과 같아지는 eps 값과 minPts 값은 각각 2.2, 4가 나왔고 이때의 Dunn index와 Silhouette index 값은 각각 0.201647값이 나왔다. 이를 바탕으로

DBSCAN을 진행하여 나온 군집화 결과는 위와 같다. 보라색, 청록색, 노란색 3개의 군집으로 군집화가 어느 정도 잘 된 것을 시각적으로 확인할 수 있다. [Q2]에서 선정한 최적 군집 개수 3이 되는 하이퍼파라미터를 이용하여 클러스터링을 진행하는 것이 아닌 하이퍼파라미터를 변화시키며 Silhouette index가 최대인 군집 수를 찾아 클러스터링을 수행한다면 더 좋은 군집화 결과를 낼 것이다.

[Q9] Noise

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

DBSCAN에서 Noise는 label이 -1로 설정된다는 점을 이용하여 다음과 같이 labels 배열에서 -1의 개수를 count하여 Noise 개수를 카운트하였다.

```
num_noise_points = np.count_nonzero(labels == -1)
```

카운트한 Noise 개수는 51개로 전체 데이터가 178개인 것을 감안하면 다소 높게 나온 것을 알 수 있다. 앞서 [Q8]의 시각화 결과에서 확인할 수 있듯이 군집 간 구분이 어려울 정도로 Noise 개수가 많은 것은 아니라고 생각한다. 하지만, Noise의 개수가 51개보다 조금 더 적으면 군집 간 구조를 더욱 명확히 파악할 수 있기에 군집의 해석이 더 용이해질 것이라고 생각한다. 이를 위해 eps의 탐색 범위를 더 증가시키고, minPts의 탐색 범위를 더 줄인다면 Noise의 개수를 줄일 수 있을 것이다. 그러나 Noise의 개수가 너무 적은 경우 의미 없는 데이터가 군집에 포함될 가능성이 증가하여 잘못된 군집화가 될 가능성이 있기에 적절한 Noise 개수를 갖도록 탐색을 진행해야 할 것이다. 또한 [Q2]의 최적의 군집 수를 갖게 하는 eps과 minPts를 찾는 것이 아닌 군집화 타당성 지표 기준 최적의 하는 eps과 minPts 탐색을 진행한다면 51개보다 적은 적절한 Noise 개수를 가질 것으로 생각한다.

[Q10] 종합

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

	K-means	Hierarchical	DBSCAN
Silhouette	0.28486	0.203787	0.201647

가장 적합한 군집화 알고리즘을 정하기에 앞서 Hierarchical의 Silhouette은 [Q6]에서 언급하였듯이 전반적인 Silhouette index가 complete의 경우가 single보다 높고, 어느 정도 군집의 세분화

(적당한 군집 수)가 되어있어야 유의미하게 군집화가 이루어져 있는 등의 이유로 complete linkage 방식으로 산출된 Silhouette값을 사용하겠다.

Silhouette 기준 가장 높은 값을 가지는 K-Meansc lustering이 가장 적합한 군집화 알고리즘이라고 생각한다. 그 뒤를 이어 Hierarchical clustering, DBSCAN clustering 순으로 적합하다고 할 수 있다. 일반적으로 Silhouette Index의 최대값을 가지는 군집 수가 최적의 군집 수로 간주되며, Dunn Index는 이를 보완하는 지표로 사용되기에 Silhouette을 기준으로 가장 적합한 알고리즘을 선정하였다.

우선 Silhouette 값이 DBSCAN이 가장 작게 나온 것은 본 데이터셋이 구형 군집화에 적합하다는 사실로 해석할 수 있다. 왜냐하면 DBSCAN은 체인 형태와 같은 비구형 군집 형성에 유리하기 때문이다. 반면 K-Means와 complete linkage hierarchical clustering의 경우 구형 군집화에 유리하기에 DBSCAN보다 성능이 좋게 나왔다고 해석할 수 있다. 동시에 이상치가 많지 않아 이상치에 강한 DBSCAN의 성능이 가장 낮게 나왔다고 해석할 수 있다. 특별히 K-Means의 경우 초기 중심점 설정 및 군집 개수 설정이 클러스터링 결과에 매우 막대한 영향을 끼치는데 가장 좋은 성능을 보이는 것으로 보아 초기 중심점 설정 및 군집 개수가 잘 설정되었다고 해석할 수 있고, 이 때문에 K-Means의 성능이 complete linkage hierarchical clustering보다 좋다고 해석할 수 있다. 이는 앞서 [Q3]에서 진행한 군집화 반복 결과를 통해서도 확인할 수 있었다.