보고서

RFM (Recency, Frequency, Monetary)

사용자**별로 얼마나 최근에, 얼마나 자주, 얼마나 많은 금액을 지출했는지에 따라** 사용자들의 분포를 확인하거나 사용자 그룹(또는 등급)을 나누어 분류하는 분석 기법

구매 가능성이 높은 고객을 선정할 때 용이한 데이터 분석방법이라고 알려져 있고, 또 사용자들의 평소 구매 패턴을 기준으로 분류를 진행하기 때문에 각 사용자 그룹의 특성에 따라 차별화된 마케팅 메시지를 전달할 수 있다.

→ 우리의 데이터도 최근 거래날짜, 가입기간 등 고객의 구매이력, 사용이력을 알 수 있기 때문에 RFM을 계산해 고객 데이터를 세분화해서 고객 그룹별 전략 인사이트를 도출할 수 있도록 한다. (우수 고객자에게 쿠폰을 더 준다는지, 구매횟수가 적거나 이탈 가능성 고객을 유지하기 위한 방안)

R : (2020-01-01 - 최근 거래 날짜)

F : 거래 횟수

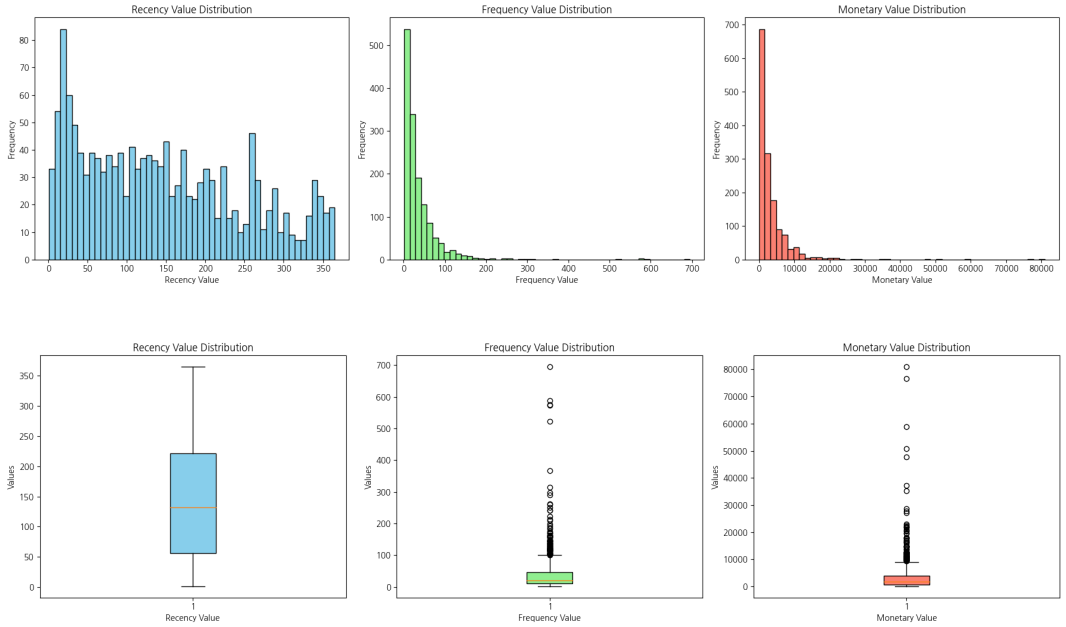
M : 수량 * 평균금액 * 할인율 * GST

- 할인율 계산 방법
    - 쿠폰 상태 여부 확인
        - 쿠폰 사용 used인 경우

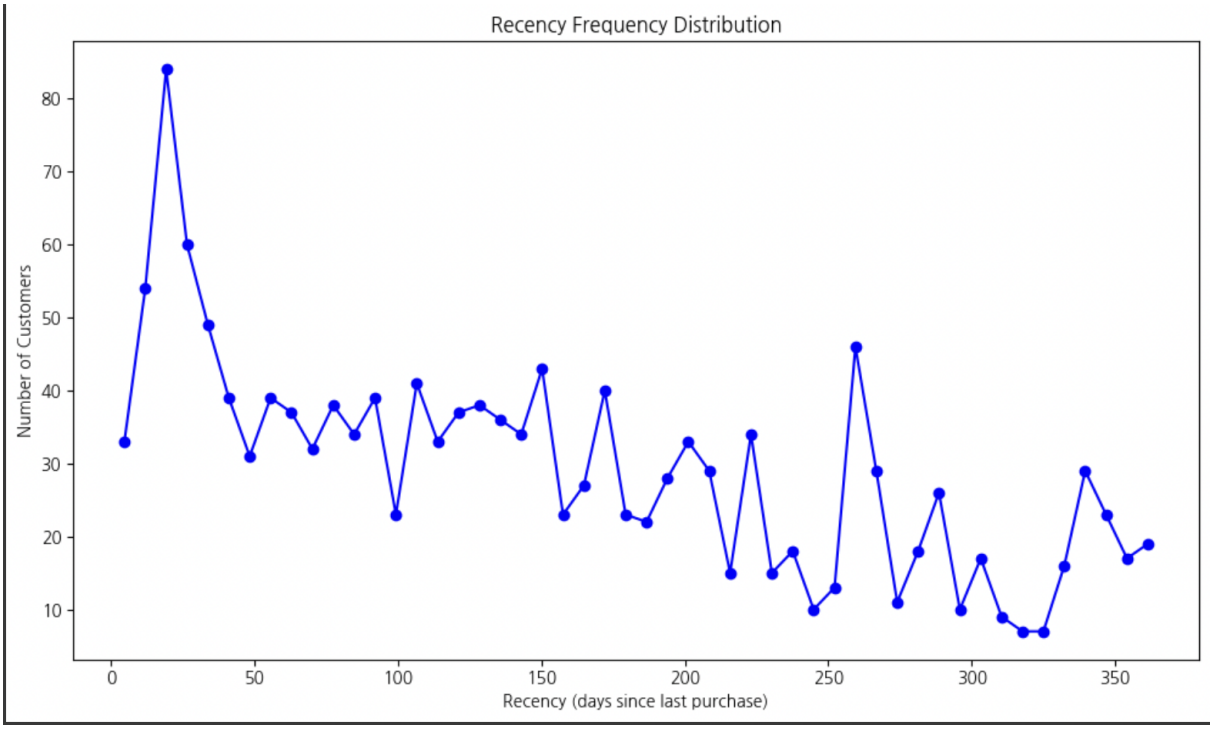
- 제품과 구매 날짜(월)에 따른 할인율 적용 (할인율 10→0.1, 20→0.2, 30→0.3) 적용

- 쿠폰 사용 not used & clicked인 경우
  - 할인율 = 1로 계산

RFM scoring 방법



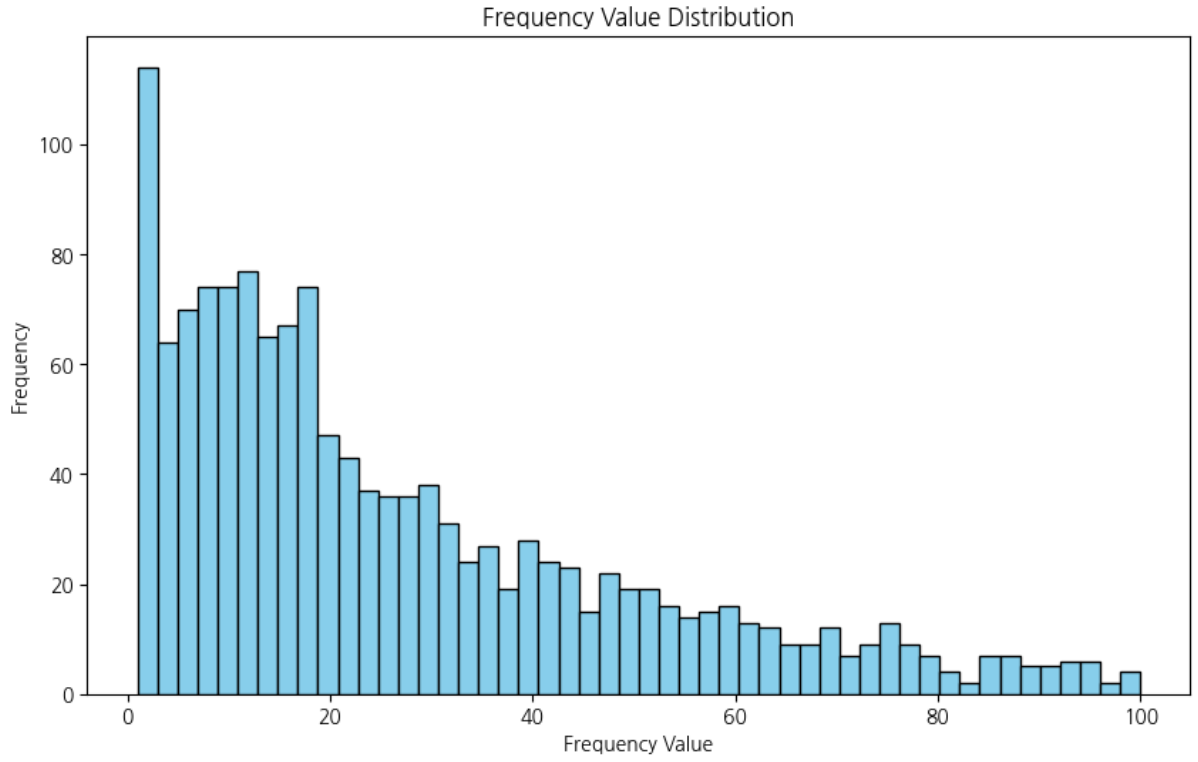
[R : 최근 거래 날짜]




```
count    1468.000000
mean      145.292234
std       101.936959
min         1.000000
25%        56.000000
50%       132.000000
75%       221.000000
max       365.000000
Name: Recency, dtype: float64
```

- 4개 구간으로 나눔 0~25%, 25~50%, 50~75%, 75~100%
- 1 > 2 > 3 > 4 (1이 가장 최근, 4가 가장 오래)

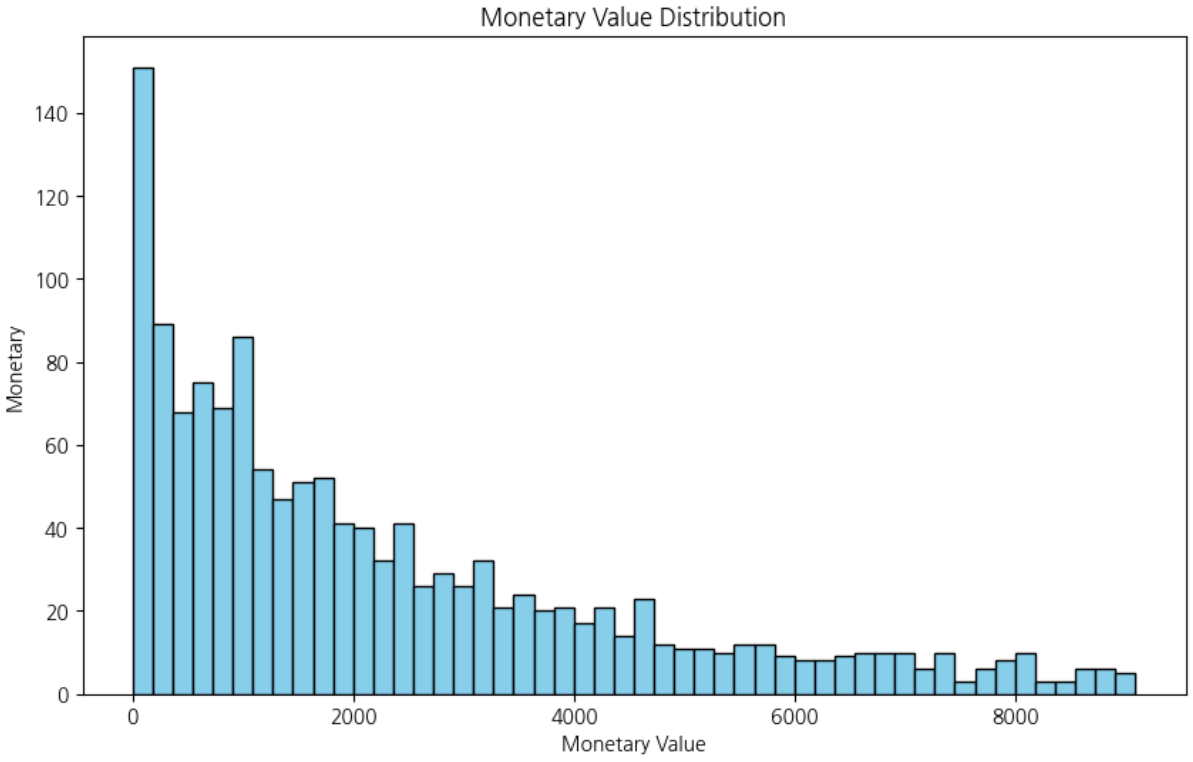
[F : 거래 횟수]



[이상치 제외한 데이터 분포]

- 최소값 ~ 25% 값 : 1
- 25% ~ 50% 값 : 2
- 50% ~ non_outlier_max값 : 3
- non_outlier_max 보다 큰 값 (이상치 값) : 4
- 1 < 2 < 3 < 4

[M : 수량*평균금액*할인율*GST]



[이상치 제외한 데이터 분포]

- 최소값 ~ 25% : 1
- 25% ~ 50% 값 : 2
- 50% ~ mon_outlier_max값 : 3
- mon_outlier_max 보다 큰 값 (이상치 값) : 4
- 1 < 2 < 3 < 4

고객 Sagement 나누기

Recency_Score : 값이 작을 수록 좋음

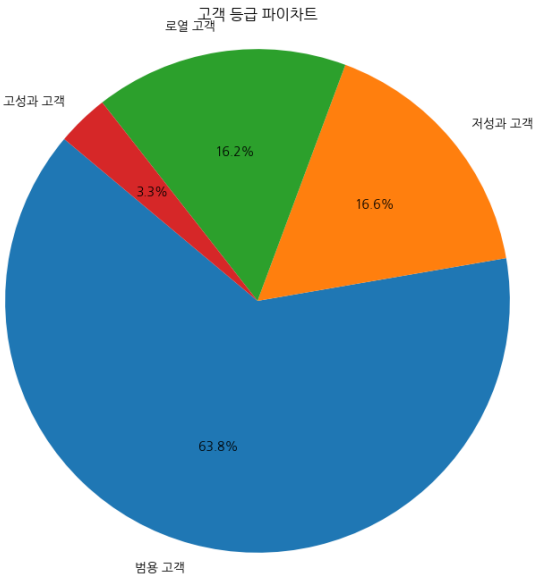
Frequency_Score, Monetary_Score : 값이 클수록 좋음

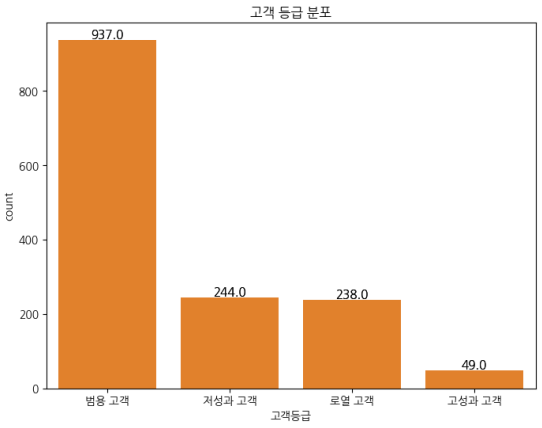
Recency_Score	Frequency_Score	Monetary_Score	고객등급
2 이하	3 이상	3 이상	로열 고객
3 이상	2 이하	3 이상	고성과 고객
그 외	그 외	그 외	범용 고객
3 이상	2 이하	2 이하	저성과 고객

```
def assign_customer_segment(row):
    if row['Recency_Score'] <=2 and row['Frequency_Score'] >=
        return '로열 고객'
    elif row['Recency_Score'] >=3 and row['Frequency_Score'] 
        return '저성과 고객'
    elif row['Recency_Score'] >=3 and row['Frequency_Score'] 
        return '고성과 고객'
    else:
        return '범용 고객'

RFM['고객등급'] = RFM.apply(assign_customer_segment, axis=1)

RFM.head(5)
```

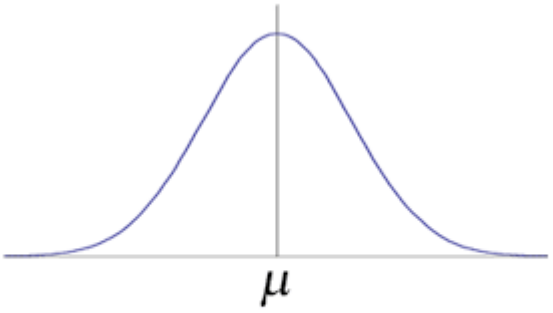




GMM (Gaussian Mixture Model)

군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것이라는 가정하에 군집화를 수행하는 방식

- 가우시안 분포 = 정규 분포



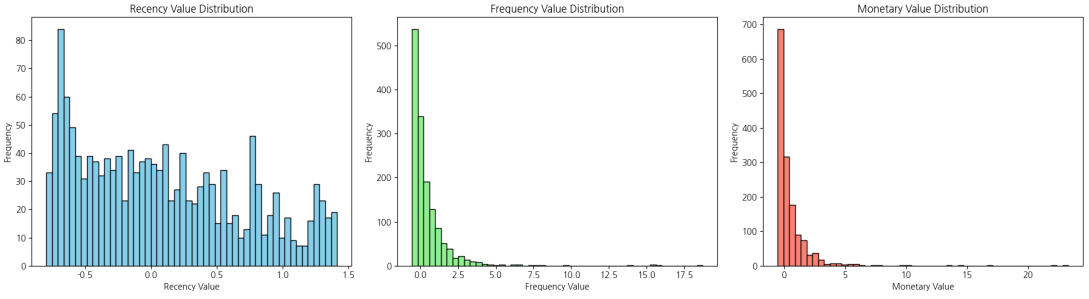
$$X \sim N(\mu, \sigma^2)$$

- 평균 μ를 중심으로 높은 데이터 분포도를 가지고 있으며, 분산이 σ^2를 따르는 분포
- 표준 정규 분포 : 표준이 0이고 표준편차가 1인 정규 분포

GMM은 데이터를 여러 개의 가우시안 분포가 섞인 것으로 간주한다. 섞인 데이터 분포에서 개별 유형의 가우시안 분포를 추출한다.

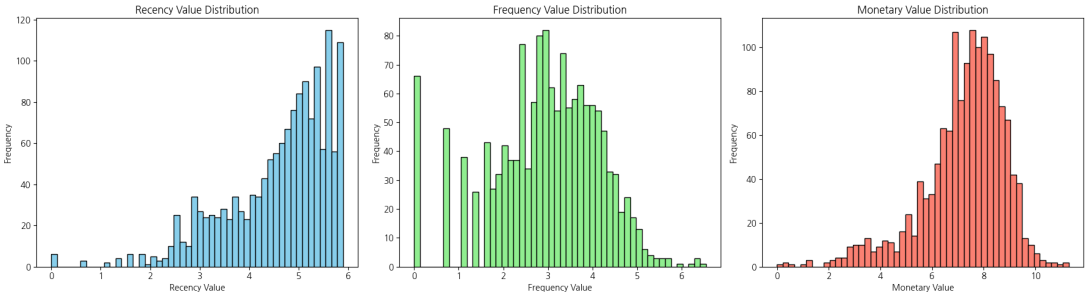
스케일링

- Robust Scaling
  - Frequency, Monetary 칼럼에 이상치 많이 존재
  - 이상치로 인해 스케일링이 왜곡될 가능성을 줄일 수 있음
  - 따라서 이상치의 영향을 줄이면서 데이터를 스케일링하는 기법이다.

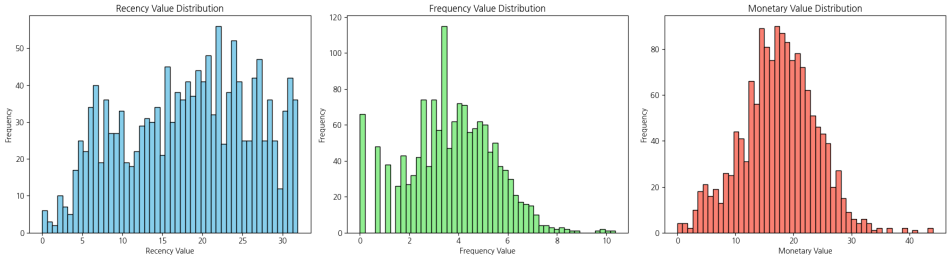


- Log Scaling
  - Frequency, Monetary 롱테일의 그래프 모습을 보임
  - 데이터의 스케일 차이를 줄이고, 큰 이상치의 영향력을 축소하는 효과가 있다.

- 데이터에 0이나 음수 값이 있는 경우, 변환 전에 작은 상수를 더해서 바꿔야 하나 우리 데이터에는 0, 음수가 없기 때문에 변환 없이 바로 사용 가능



- Box-Cox 변환
  - 데이터 모두 양수인 경우에만 적용 가능
  - 정규 분포에 가깝게 만드는 데 도움을 줄 수 있으며, 로그 변환보다 더 일반적인 형태를 가지고 있다.
  - lambda 값을 조절하면서 분포가 정규분포가 되는 최적의 lambda 값 탐색 (데이터 분산 안정화)



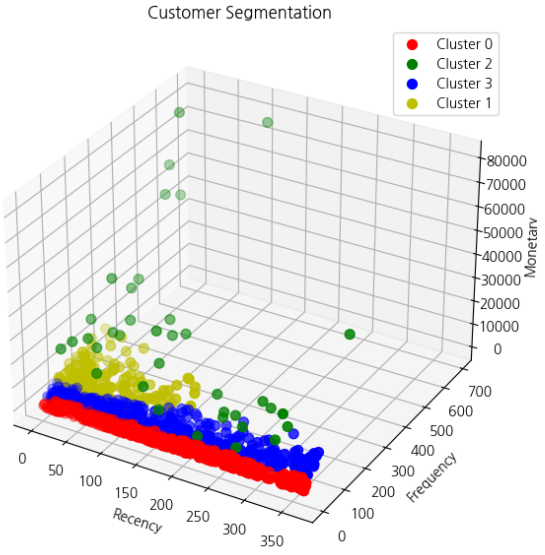
모델 학습 - 군집화 4개

‘로열고객’ : 최근 거래했고, 거래 빈도 잦고, 1년동안 총 구매 금액이 많은 고객

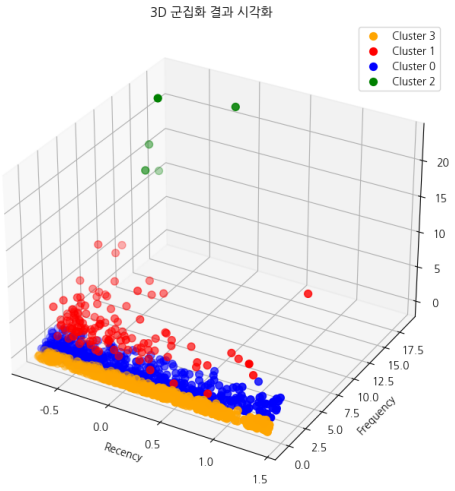
‘고성과 고객’ : 최근 거래보단 , 거래 빈도가 낮고, 1년동안 총 구매 금액이 많은 고객

‘저성과 고객’ : 최근 거래 하지 않고, 거래도 잘 하지 않고, 1년동안 총 구매 금액도 적은 고객

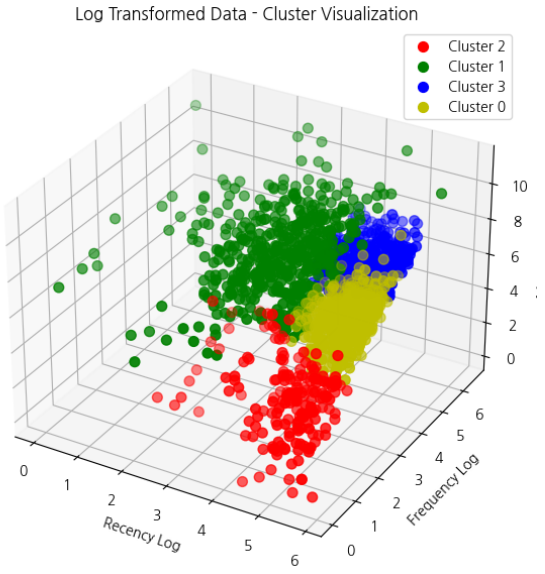
‘범용 고객’ : 일반적인 고객



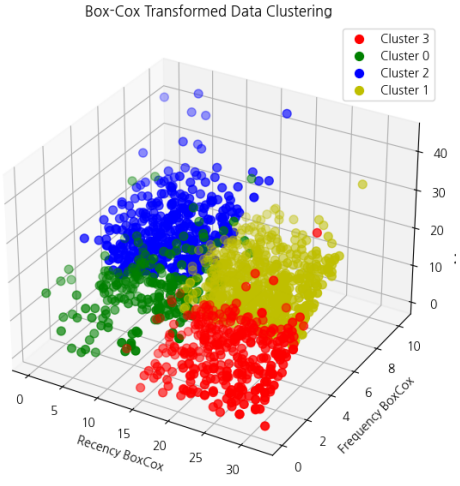
스케일링 X



Robust



Log



Box-Cox

스케일링을 하지 않았을 때와 Robust 스케일링을 적용했을 때, 군집 분석 결과에서 Recency 값의 영향력이 크게 감소한 것으로 보인다. 고객 데이터 분석에서 Recency는 중요한 요소 중 하나이므로, 이를 충분히 반영하지 못한 것은 바람직하지 않아 보인다.

Log, Box-Cox 변환 후, 군집 분석 결과는 Cluster 구분이 잘 된 것으로 보인다.

[Log의 경우]

- Cluster 2 : Recency (중앙~높은 값), Frequency (작은~중앙값), Moneatary 낮은 값 ➡️ 저성과 고객

- Cluster 1 : Recency (중앙값), Frequency (중앙), Monetary(중앙) ➡️ 범용 고객
- Cluster 3 : Recency (높은값), Frequency (높은값), Monetary(중앙~높은값)
- Cluster 0 : Recency (중앙~높은값), Frequency (중앙~높은값), Monetary(낮은~중앙값)

→ 명확하게 4개의 Cluster 값을 '로열 고객', '고성과 고객', '저성과 고객', '범용고객'으로 나누기 힘들어 보임 (고성과, 로열 고객 나누기 애매해다.)

[Box-Cox]

- Cluster 3 : Recency (중앙~높은 값), Frequency (작은~중앙값), Moneatary 낮은 값 ➡️ 저성과 고객
- Cluster 0 : Recency (낮은~중앙값), Frequency (골고루), Monetary(낮은~중앙값) ➡️ 범용 고객
- Cluster 2 : Recency (낮은~중앙값), Frequency (중앙~높은값), Monetary(중앙~높은값) ➡️ 로열 고객
- Cluster 1 : Recency (중앙~높은값), Frequency (중앙~높은값), Monetary(낮은~중앙값) ➡️ 고성과 고객

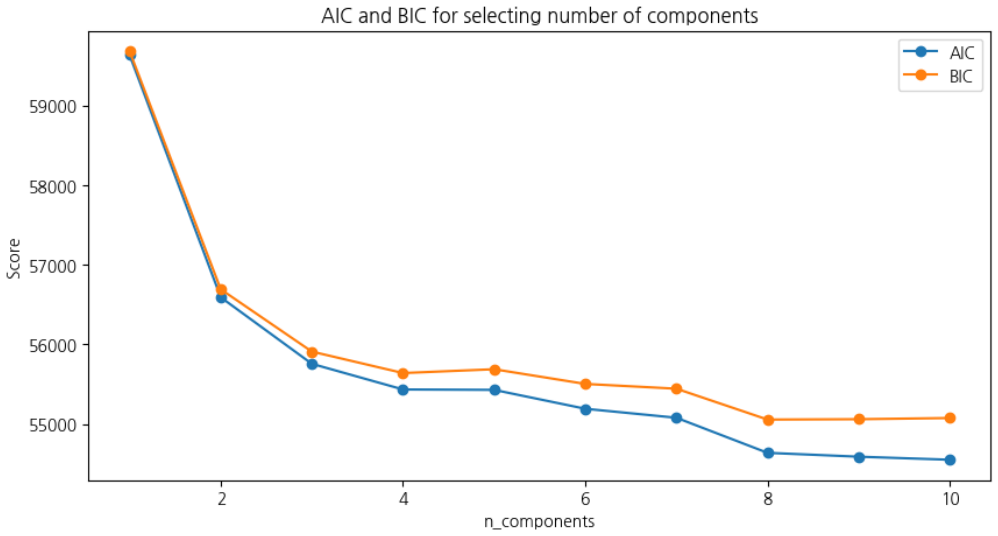
모델 학습 - 군집화 10개

최적 군집수 구하기

GMM 모델 학습에 군집화 수 4개가 적절하지 않을 수 있어서, 최적의 군집수 구함

AIC, BIC 방법을 이용해 군집수 구하기

- AIC, BIC 모두 모델의 복잡성에 대한 벌점을 포함하며, 값이 낮을수록 모델이 데이터에 더 잘 맞는다고 평가한다.
- 모델이 데이터를 얼마나 잘 설명하는지와 군집의 수가 많아질수록 증가하는 모델의 복잡도 사이의 균형을 찾는 데 사용된다.



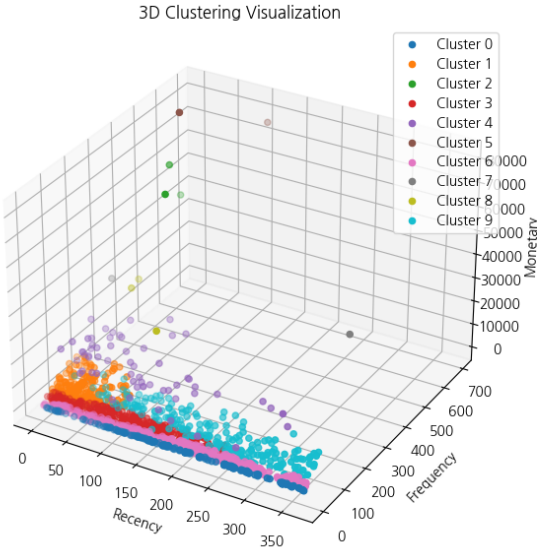
[AIC]

- AIC는 주어진 데이터셋에 대해 상대적으로 더 좋은 모델을 선택하는 데 도움을 준다.
- 모델의 적합도를 강조하며, 때로는 복잡한 모델을 선호할 수 있다.
- 데이터에 대한 모델의 설명력을 최대화하는 데 초점을 맞춘다.

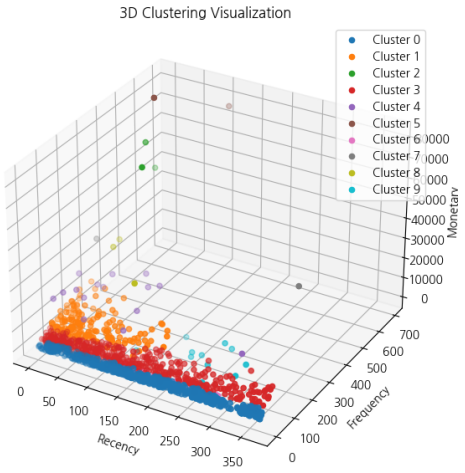
[BIC]

- BIC는 샘플 크기가 클 때 AIC보다 더 많은 벌점을 부여하여 모델의 복잡성을 더 강하게 패널티를 준다.
- 일반적으로 더 단순한 모델을 선호하며, 군집 수가 많은 경우 그에 대한 더 큰 패널티를 부여한다.
- 모델 선택에서 과적합(overfitting)을 방지하는 데 더 효과적일 수 있다.

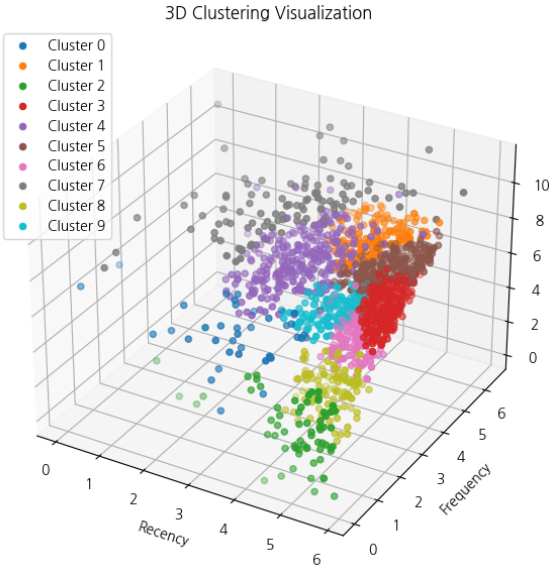
⇒ 군집화를 수행하는 주된 목적이 “모델이 데이터를 얼마나 잘 설명하는지”를 극대화하는 것이기에 BIC 결과 따르기로 함



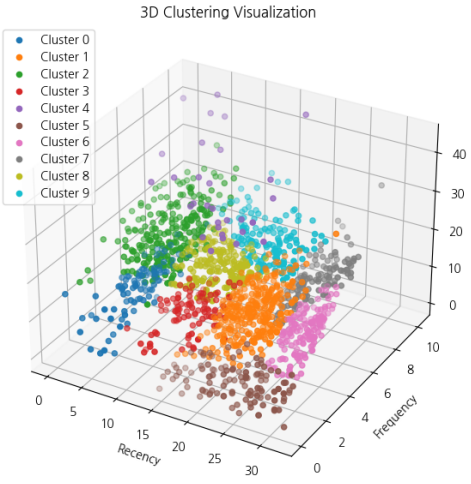
scaling X



Robust



Log



Box-Cox

→ 군집화를 10개로 나눌 경우, 데이터를 Box-Cox 변환후 군집화 한 모델 결과가 가장 적합해보인다.