실루엣 계수를 이용한 군집분석 품질 평가

# 실루엣 계수란?

실루엣 계수(Silhouette Score)는 군집분석의 품질을 평가하는 지표로, 개별 데이터 포인트가 잘 군집화되었는지를 평가합니다. 실루엣 계수는 -1에서 1 사이의 값을 가지며, 값이 클수록 군집화가 잘된 것을 의미합니다.

# 실루엣 계수 계산 방법

실루엣 계수 s(i)는 다음과 같이 계산됩니다:  
1. a(i): 동일 군집 내의 평균 거리  
- 데이터 포인트 i와 동일 군집에 속한 다른 모든 데이터 포인트들 사이의 평균 거리입니다.  
- a(i) 값이 작을수록 같은 군집 내의 데이터 포인트들이 서로 가깝다는 것을 의미합니다.  
2. b(i): 가장 가까운 다른 군집까지의 평균 거리  
- 데이터 포인트 i와 가장 가까운 다른 군집에 속한 모든 데이터 포인트들 사이의 평균 거리입니다.  
- b(i) 값이 클수록 다른 군집과 잘 분리되어 있다는 것을 의미합니다.  
3. 실루엣 계수 계산  
- 실루엣 계수 s(i)는 다음과 같이 계산됩니다:  
s(i) = (b(i) - a(i)) / max(a(i), b(i))  
  
- s(i) 값의 범위:  
 - s(i)가 1에 가까울수록 데이터 포인트가 잘 군집화되어 있음.  
 - s(i)가 0에 가까울수록 데이터 포인트가 경계에 위치해 있음.  
 - s(i)가 -1에 가까울수록 데이터 포인트가 잘못 군집화되어 있음.

# 군집 전체의 실루엣 계수

군집 전체의 실루엣 계수는 개별 데이터 포인트들의 실루엣 계수의 평균으로 계산됩니다. 이 값을 통해 군집화의 전체적인 품질을 평가할 수 있습니다.

# 실루엣 계수를 이용한 최적의 군집 수 찾기

1. 데이터 준비 및 PCA 처리  
2. 엘보우 방법을 통한 군집의 수 결정  
3. 실루엣 계수를 통한 군집 품질 평가  
4. 군집분석 수행 및 실루엣 계수 계산

# 파이썬 코드 예시

### 데이터 준비 및 PCA 처리  
```python  
import pandas as pd  
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA  
from sklearn.cluster import KMeans  
from sklearn.metrics import silhouette\_score  
import matplotlib.pyplot as plt  
  
# 데이터 로드 및 선택  
data = pd.read\_excel('/mnt/data/행복요인\_data.xlsx')  
data\_selected = data[['경제1', '경제2', '경제3', '지식1', '지식2', '지식3', '문화1', '문화2', '문화3', '사회1', '사회2', '사회3', '정서1', '정서2', '정서3', '행복1', '행복2', '행복3']]  
  
# 데이터 스케일링  
scaler = StandardScaler()  
data\_scaled = scaler.fit\_transform(data\_selected)  
  
# PCA 처리  
pca = PCA(n\_components=5)  
data\_pca = pca.fit\_transform(data\_scaled)  
```  
  
### 최적의 군집 수 찾기  
```python  
silhouette\_scores = []  
for k in range(2, 11):  
 kmeans = KMeans(n\_clusters=k, random\_state=42)  
 kmeans.fit(data\_pca)  
 score = silhouette\_score(data\_pca, kmeans.labels\_)  
 silhouette\_scores.append(score)  
  
plt.figure(figsize=(10, 6))  
plt.plot(range(2, 11), silhouette\_scores, marker='o')  
plt.title('Silhouette Scores for Optimal k')  
plt.xlabel('Number of clusters')  
plt.ylabel('Silhouette Score')  
plt.show()  
```  
  
### 군집분석 수행 및 실루엣 계수 계산  
```python  
# 최적의 군집 수로 KMeans 수행 (예: 실루엣 계수가 가장 높은 군집 수 선택)  
optimal\_k = silhouette\_scores.index(max(silhouette\_scores)) + 2 # range 시작이 2이므로 +2  
kmeans = KMeans(n\_clusters=optimal\_k, random\_state=42)  
kmeans.fit(data\_pca)  
labels = kmeans.labels\_  
  
# 군집의 실루엣 계수 계산  
silhouette\_avg = silhouette\_score(data\_pca, labels)  
print(f'For n\_clusters = {optimal\_k}, the average silhouette\_score is : {silhouette\_avg}')  
```

# 결과 해석

- 실루엣 계수가 1에 가까울수록 데이터 포인트들이 잘 군집화된 것입니다.  
- 군집의 수가 증가하면 실루엣 계수가 증가할 수도 있지만, 군집의 수가 너무 많아지면 과적합이 발생할 수 있습니다.  
- 최적의 군집 수를 선택할 때는 실루엣 계수와 함께 해석의 용이성도 고려해야 합니다.