

강원지역혁신플랫폼

기계학습

Machine Learning

K-평균 군집 실습(1)



▼ 학습목표

- ▶ 백화점 고객 샘플 데이터 집합으로
군집 분석을 구현할 수 있습니다.





01 | 실습

❖ (권장) 아래와 같은 경로에 실행 소스가 존재하면 환경 구축 완료

- ◆ 구글 드라이브 “PyWork > ML” 폴더로 이동함
 - 아래의 [ch10] 폴더를 클릭하면 됨

내 드라이브 > PyWork > ML

유형 ▾ 사람 ▾ 수정 날짜 ▾

이름	▲
ch09	
ch10	
ch11	
ch12	
ch13	
ch14	
HelloWorld	



01 | 실습

- ◆ “ML > ch10 >” 폴더를 클릭함
 - › 아래의 [ch10_01_k-평균군집분석실습.ipynb] 스크립트를 클릭함

... > ML > ch10 ▾

유형 ▾ 사람 ▾ 수정 날짜 ▾

이름 ↑

- ch10_01_k-평균군집분석실습.ipynb
- ch10_02_k-평균군집분석실습.ipynb
- department_store_customers.txt
- department_store_customers.xlsx
- mall_customers_data.txt
- mall_customers_data.xlsx



02 | 비지도 학습: K-평균 군집 분석



백화점 고객 샘플 데이터로 타겟(target) 고객 분류하기

△ 다음은 백화점 고객 샘플 데이터 500개로 마케팅을 위한 타겟 고객 분류를 수행해 보자.

◆ 백화점 고객 샘플 데이터 집합은 다음과 같음

› 백화점 고객 샘플 데이터 집합은 500개의 관측치와 7개의 속성으로 구성됨

No	속성	속성 설명
1	이름	고객 이름
2	나이	고객 나이
3	성별	성별(여성, 남성)
4	직업	직업(조경 기술자, 총무 사무원, 항공기 조종사, 등)
5	구매액	상품을 구매한 금액
6	구매일자	상품을 구매한 일자
7	거주지역	서울시 거주지역(구)



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 백화점 샘플 데이터 집합을 읽어오는 코드이다.

- ◆ 실행 결과 데이터의 형상이 (500, 7)인 것을 볼 수 있음
 - 관측치가 500개, 7개의 속성으로 구성됨

```
df = pd.read_excel(os.getcwd()+'/department_store_customers.xlsx')  
print(df.shape) # (500, 7)
```



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 백화점 샘플 데이터 집합을 앞에서 5개 행을 출력하는 코드이다.

◆ 실행 결과 아래와 같은 데이터로 구성된 것을 볼 수 있음

```
df.head()
```

	이름	나이	성별	직업	구매액	구매일자	거주지역
0	김지후	49	여성	기타 행정 및 경영지원 관리자	518441	2023-09-07	송파구
1	윤성현	75	여성	경영지원 관리자	706071	2024-02-23	관악구
2	손성희	19	남성	임상 심리사 및 기타 치료사	473153	2024-03-15	서초구
3	오정숙	56	남성	건설 및 광업 단순 종사원	244905	2023-07-06	종로구
4	강채원	72	남성	물품 이동 장비 조작원	714430	2024-01-15	노원구



02 | 비지도 학습: K-평균 군집 분석

▲ 다음은 백화점 샘플 데이터 집합의 “나이”, “성별”, “직업”, “구매액”, “거주지역” 속성으로 K-평균 군집 분석을 수행해 보자.

- ◆ 백화점 샘플 데이터 집합에서 “나이”, “성별”, “직업”, “구매액”, “거주지역” 속성을 선택하여 변수 X에 할당함

```
X = df[['나이', '성별', '직업', '구매액', '거주지역']]
```



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 범주형 변수 “성별”, “거주지역”, “직업”의 데이터를 숫자형으로 변환하는 코드이다.

- ◆ 예를 들어 성별의 경우 “여성” → 1, “남성” → 0으로 변환됨

```
label_encoder = LabelEncoder()  
X['성별'] = label_encoder.fit_transform(X['성별'])  
X['거주지역'] = label_encoder.fit_transform(X['거주지역'])  
X['직업'] = label_encoder.fit_transform(X['직업'])
```



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 표준화 함수를 이용한 데이터 정규화를 수행하는 코드이다.

◆ 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()  
X_scaled = scaler.fit_transform(X)  
X_scaled
```

	나이	성별	직업	구매액	거주지역
0	49	1	76	518441	17
1	75	1	22	706071	4
2	19	0	196	473153	14
3	56	0	10	244905	22
4	72	0	100	714430	8

정규화 이전



```
array( [[-0.02092837,  1.04920939, -0.88079422,  0.02289189,  0.78927396],  
       [ 1.42624634,  1.04920939, -1.52035543,  0.66213514, -1.0216264 ],  
       [-1.69074535, -0.9530986 ,  0.54045291, -0.13140139,  0.37137387],  
       ...  
       [-1.52376365,  1.04920939,  0.78917116, -0.36383613, -0.74302635],  
       [-1.24546082,  1.04920939, -1.31901209,  0.44828522, -1.0216264 ],  
       [ 1.42624634,  1.04920939, -1.31901209, -0.25314821, -0.32512626]] )
```

정규화 이후



02 | 비지도 학습: K-평균 군집 분석

△ 다음은 엘보우(elbow)방법을 이용해 최적의 K값을 찾는 코드이다.

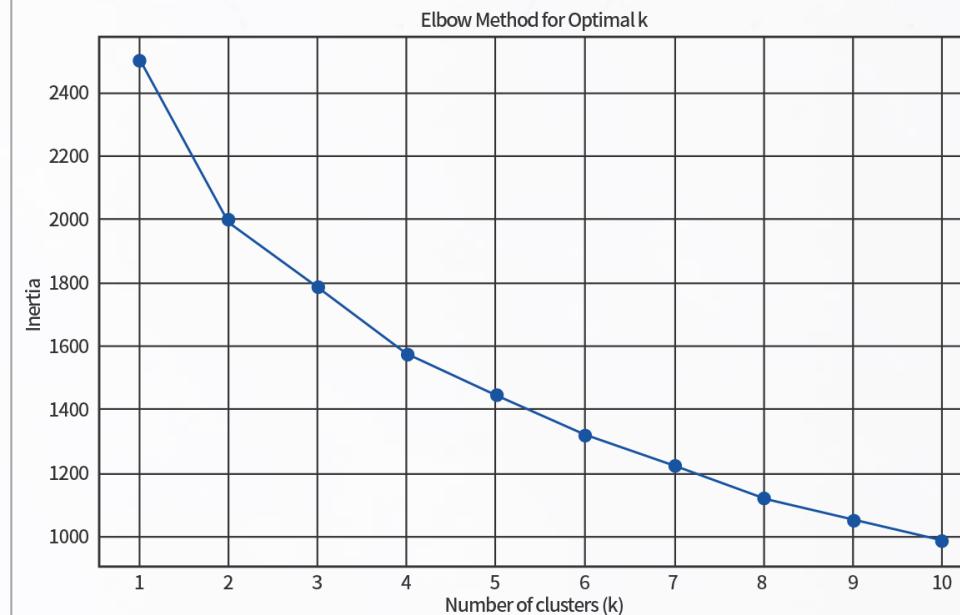
◆ 최적의 K값을 찾기 위해 $K=1\sim10$ 까지 범위로 오차 제곱합(SSE)의 변화를 시각화함

› 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
# 엘보우 방법을 사용하여 최적의 k 값 찾기
inertia = []
k_range = range(1, 11)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# 그래프로 그리기
plt.figure(figsize=(10, 6))
plt.plot(k_range, inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_range)
plt.grid(True)
plt.show()
```



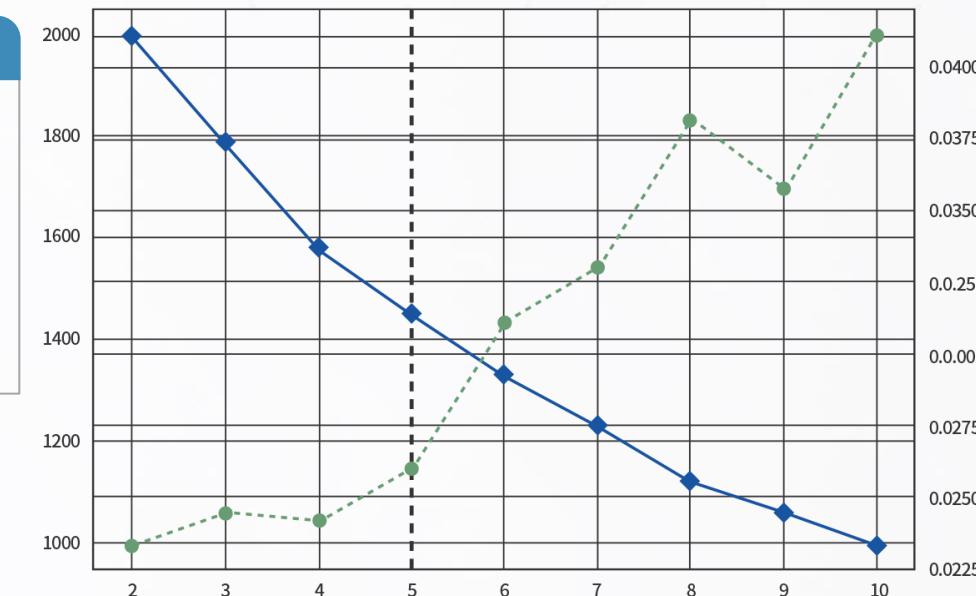


02 | 비지도 학습: K-평균 군집 분석

△ 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.

- ◆ 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
 - › 검정색 점선의 위치를 보았을 때, 여기서는 K=5인 경우를 추천해주고 있음
 - › 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer  
  
model = KMeans()  
visualizer = KElbowVisualizer(model, k=(2,11))  
visualizer.fit(X_scaled)
```





02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 최적의 K값을 5로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.

◆ 아래와 같이 “나이”, “성별”, “직업”, “구매액”, “거주지역” 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=5, random_state=42)  
kmeans.fit_predict(X_scaled)  
  
# 각 군집의 중심값 확인  
centroids = kmeans.cluster_centers_  
centroids_df = pd.DataFrame(centroids, columns=X.columns)  
centroids_df
```

	나이	성별	직업	구매액	거주지역
0	0.241004	1.049209	0.031521	-0.781636	0.132720
1	0.897471	-0.928680	0.723886	-0.020509	0.209990
2	-0.319524	1.049209	0.036971	0.974726	-0.313518
3	-0.679680	-0.843082	-0.341319	0.137348	1.011236
4	-0.145725	-0.953099	-0.380366	0.020124	-1.023093



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 최적의 K값을 5로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.

◆ 아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=5, random_state=42)  
X['군집'] = kmeans.fit_predict(X_scaled)  
df['군집'] = kmeans.fit_predict(X_scaled)
```

	나이	성별	직업	구매액	거주지역	군집
0	49	1	76	518441	17	0
1	75	1	22	706071	4	2
2	19	0	196	473153	14	3
3	56	0	10	244905	22	3
4	72	0	100	714430	8	4

X 변수

	이름	나이	성별	직업	구매액	구매일자	거주지역	군집
0	김지후	49	여성	기타 행정 및 경영지원 관리자	518441	2023-09-07	송파구	0
1	윤성현	75	여성	경영지원 관리자	706071	2024-02-23	관악구	2
2	손성희	19	남성	임상 심리사 및 기타 치료사	473153	2024-03-15	서초구	3
3	오정숙	56	남성	건설 및 광업 단순 종사원	244905	2023-07-06	종로구	3
4	강채원	72	남성	물품 이동 장비 조작원	714430	2024-01-15	노원구	4

df 변수



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 군집별 고객 수를 확인하는 코드이다.

◆ 아래와 같이 5개 군집의 고객 수를 확인할 수 있음

```
cluster_counts = df['군집'].value_counts()  
print(cluster_counts)
```

군집

0	136
2	96
4	95
3	91
1	82

Name: count, dtype: int64

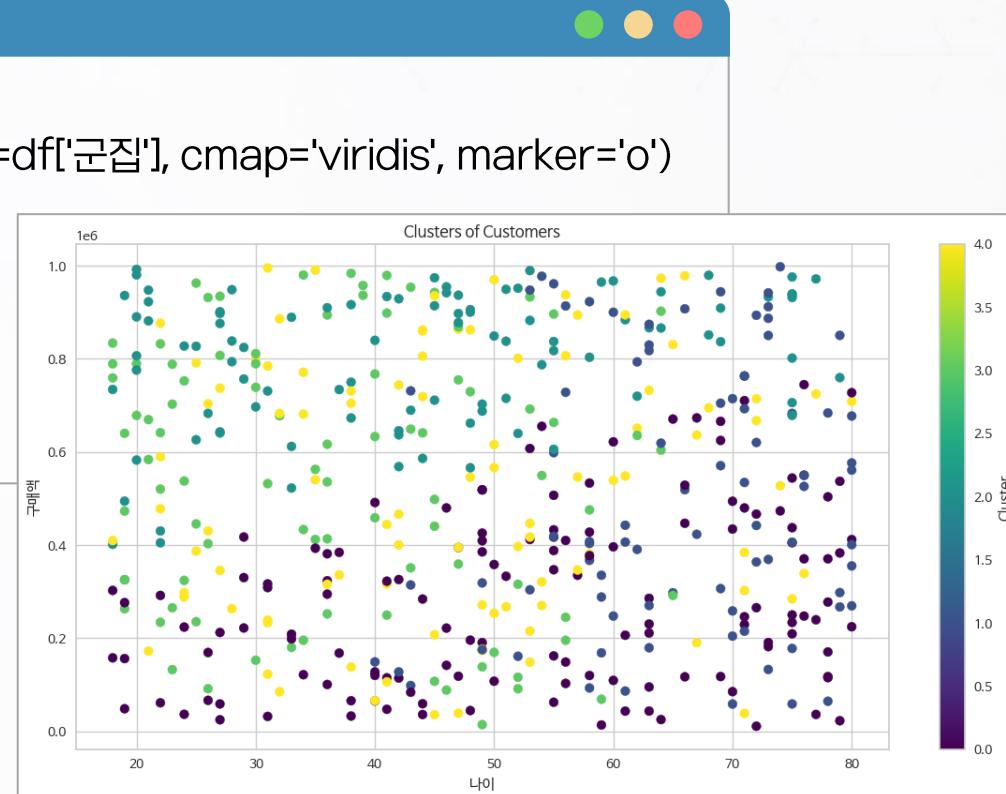


02 | 비지도 학습: K-평균 군집 분석

△ 다음은 군집별 시각화를 구현하는 코드이다.

- ◆ 아래 그림은 “나이”, “성별”, “직업”, “구매액”, “거주지역” 속성, K=5 값으로 K-평균 군집 결과 “나이”와 “구매액” 군집별 시각화 결과임
- 시각화 결과에서 **5개 군집으로 잘 분류되지 않는 것**을 볼 수 있음

```
plt.figure(figsize=(14, 7))
plt.scatter(df['나이'], df['구매액'], c=df['군집'], cmap='viridis', marker='o')
plt.title('Clusters of Customers')
plt.xlabel('나이')
plt.ylabel('구매액')
plt.colorbar(label='Cluster')
plt.show()
```





02 | 비지도 학습: K-평균 군집 분석

- ▲ 다음은 백화점 샘플 데이터 집합의 “나이”, “성별”, “구매액” 속성으로 K-평균 군집 분석을 수행해 보자.
- ◆ 백화점 샘플 데이터 집합에서 “나이”, “성별”, “구매액” 속성을 선택하여 변수 df_selected에 할당함

```
# 필요한 속성 선택  
df_selected = df[['나이', '성별', '구매액']]  
  
df_selected
```

	나이	성별	구매액
0	49	여성	518441
1	75	여성	706071
2	19	남성	473153
3	56	남성	244905
4	72	남성	714430
...
495	55	여성	161926



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 범주형 변수 “성별”을 수치형 변수로 변환하는 코드이다.

◆ 예를 들어 성별의 경우 “여성” → 1, “남성” → 0으로 변환됨

```
le = LabelEncoder()  
df_selected['성별'] = le.fit_transform(df_selected['성별'])  
  
df_selected
```

	나이	성별	구매액
0	49	여성	518441
1	75	여성	706071
2	19	남성	473153
3	56	남성	244905
4	72	남성	714430
...
495	55	여성	161926



	나이	성별	구매액
0	49	1	518441
1	75	1	706071
2	19	0	473153
3	56	0	244905
4	72	0	714430
...
495	55	1	161926

변경 이전

변경 이후

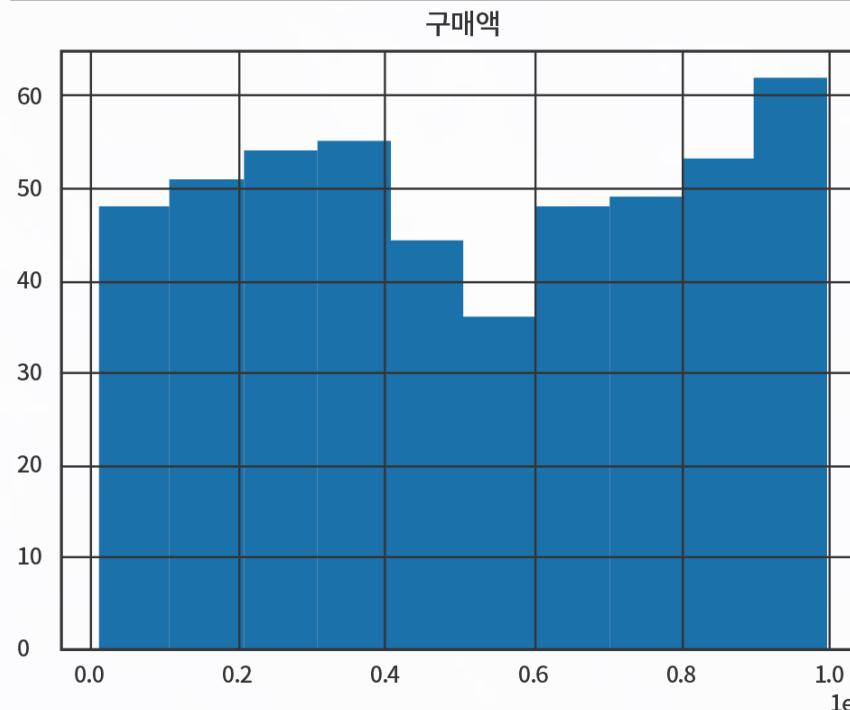


02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 “구매액” 변수로 **히스토그램**을 그리는 코드이다.

◆ 아래의 히스토그램 그래프와 같이 구매액은 **0 ~ 1,000,000 범위**에 있음

```
pd.DataFrame(df_selected, columns=['구매액']).hist()  
plt.subplots_adjust(hspace=1)  
plt.show()
```





02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 표준화 함수로 데이터 정규화를 수행하는 코드이다.

◆ 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()  
df_scaled = scaler.fit_transform(df_selected)  
df_scaled
```

```
array([[-0.02092837,  1.04920939,  0.02289189],  
       [ 1.42624634,  1.04920939,  0.66213514],  
      [-1.69074535, -0.9530986 , -0.13140139],  
      ...,  
      [-1.52376365,  1.04920939, -0.36383613],  
      [-1.24546082,  1.04920939,  0.44828522],  
      [ 1.42624634,  1.04920939, -0.25314821]])
```



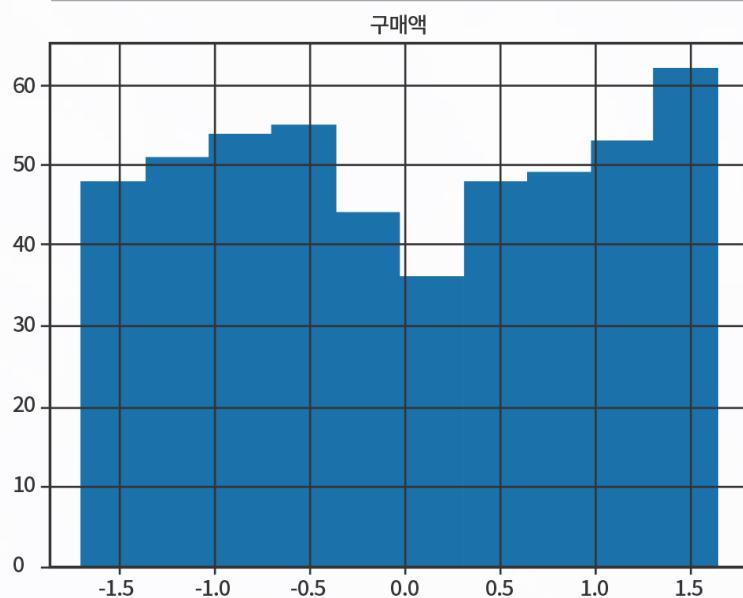
02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 데이터 정규화된 “구매액” 변수로 히스토그램을 그리는 코드이다.

◆ 아래의 히스토그램 그래프와 같이 구매액은 약 -1.5 ~ 1.5 범위에 있음

```
df_scaled_data = pd.DataFrame(df_scaled, columns=['나이','성별','구매액'])

pd.DataFrame(df_scaled_data, columns=['구매액']).hist()
plt.subplots_adjust(hspace=1)
plt.show()
```





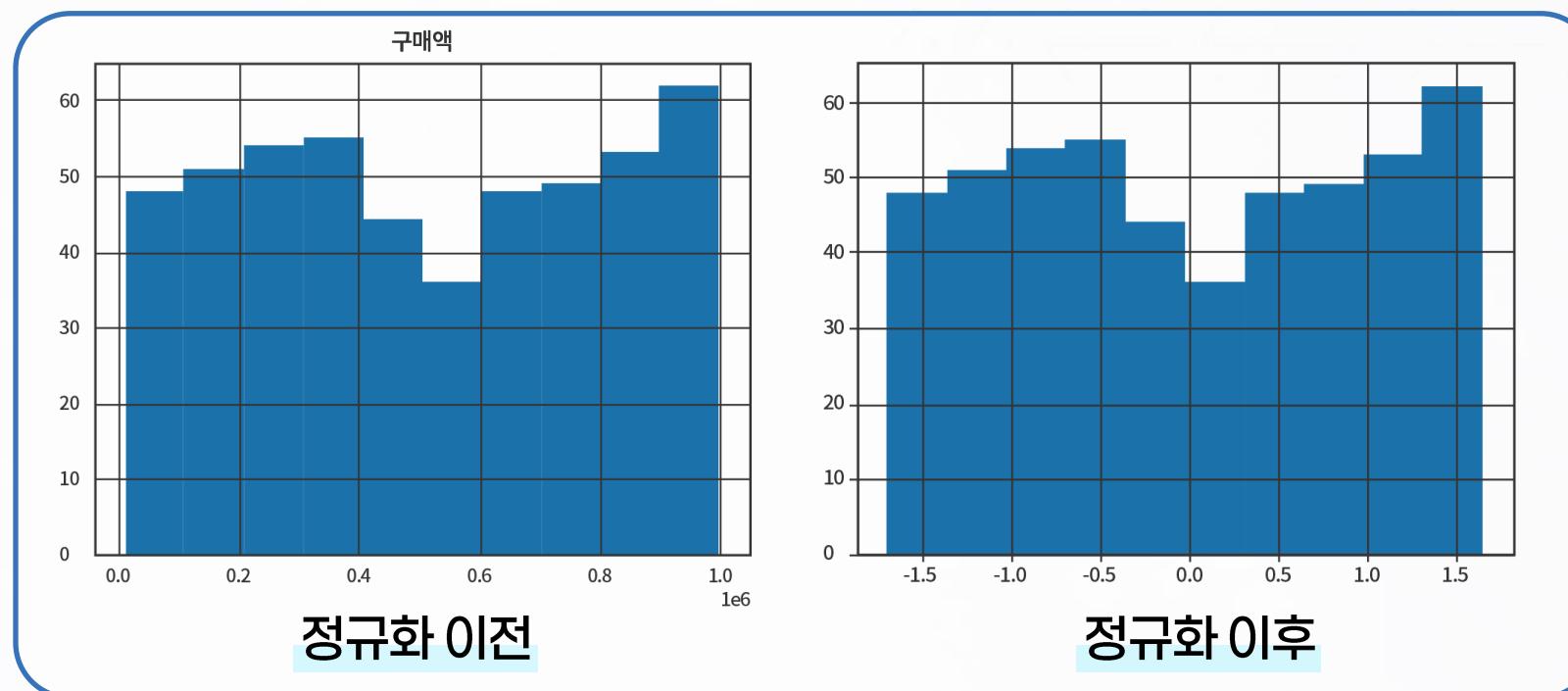
02 | 비지도 학습: K-평균 군집 분석

아래의 그림은 “구매액” 변수의 히스토그램을 정규화 이전과 이후를 비교한 것임

◆ 아래 그래프와 같이 데이터의 규격이 달라진 것을 알 수 있음

› 정규화 이전 구매액: 0 ~ 1,000,000 범위의 값

› 정규화 이후 구매액: 약 -1.5 ~ 1.5 범위의 값





02 | 비지도 학습: K-평균 군집 분석

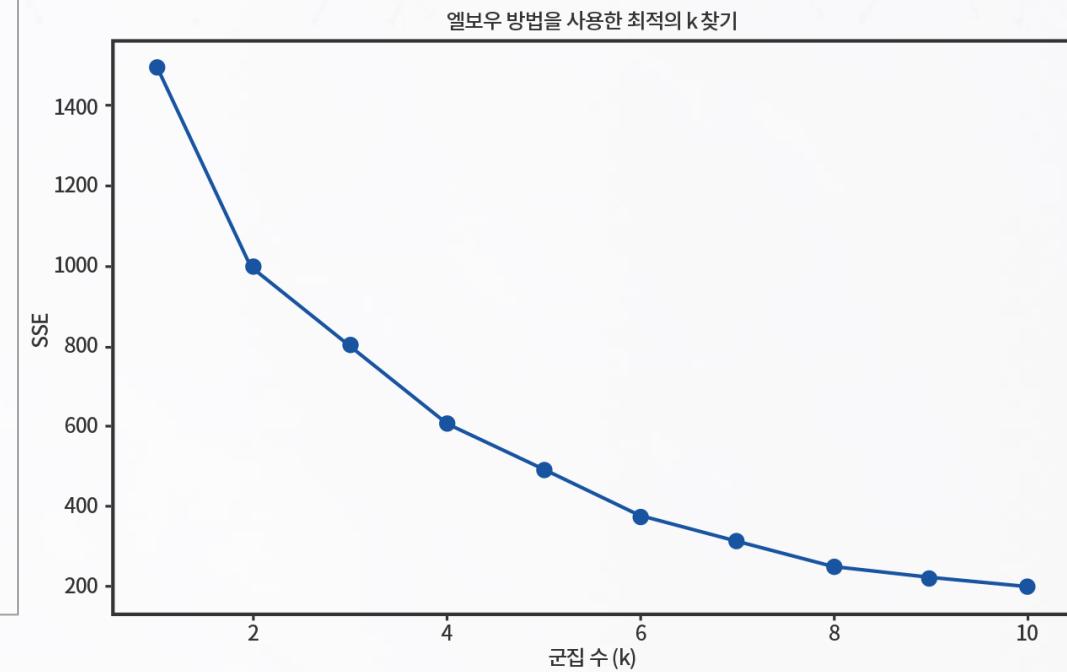
△ 다음은 엘보우(elbow) 방법을 이용해 최적의 K값을 찾는 코드이다.

◆ 최적의 K값을 찾기 위해 K=1~10까지 범위로 오차 제곱합(SSE)의 변화를 시각화함

› 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_scaled)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse, marker='o')
plt.title('엘보우 방법을 사용한 최적의 k 찾기')
plt.xlabel('군집 수 (k)')
plt.ylabel('SSE')
plt.show()
```





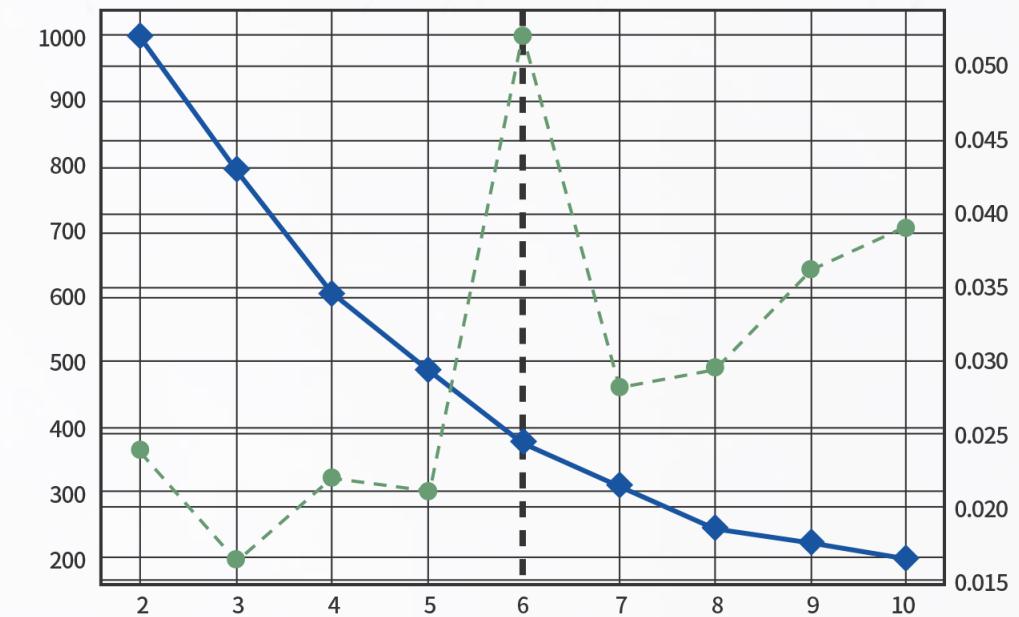
02 | 비지도 학습: K-평균 군집 분석

△ 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.

- ◆ 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
 - > 검정색 점선의 위치를 보았을 때, 여기서는 K=6인 경우를 추천해주고 있음
 - > 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,11))
visualizer.fit(df_scaled)
```





02 | 비지도 학습: K-평균 군집 분석

- ❖ 다음은 최적의 K값을 6으로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.
- ◆ 아래와 같이 “나이”, “성별”, “구매액” 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 6으로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=6, random_state=42)  
kmeans.fit_predict(df_scaled)  
  
# 각 군집의 중심값 확인  
centroids = kmeans.cluster_centers_  
centroids_df = pd.DataFrame(centroids,  
columns=df_selected.columns)  
centroids_df
```

	나이	성별	구매액
0	-0.299836	1.049209	1.025688
1	1.011064	-0.953099	0.064195
2	-0.886276	1.049209	-0.938307
3	-0.528500	-0.953099	-0.893235
4	1.007435	1.049209	-0.544171
5	-0.662416	-0.953099	0.988138



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 최적의 K값을 6으로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.

◆ 아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 6으로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=6, random_state=42)  
df['군집'] = kmeans.fit_predict(df_scaled)  
df_selected['군집'] = kmeans.fit_predict(df_scaled)
```

	이름	나이	성별	직업	구매액	구매일자	거주지역	군집
0	김지후	49	여성	기타 행정 및 경영지원 관리자	518441	2023-09-07	송파구	0
1	윤성현	75	여성	경영지원 관리자	706071	2024-02-23	관악구	4
2	손성희	19	남성	임상 심리사 및 기타 치료사	473153	2024-03-15	서초구	3
3	오정숙	56	남성	건설 및 광업 단순 종사원	244905	2023-07-06	종로구	3
4	강채원	72	남성	물품 이동 장비 조작원	714430	2024-01-15	노원구	1

df 변수

	나이	성별	구매액	군집
0	49	1	518441	0
1	75	1	706071	4
2	19	0	473153	3
3	56	0	244905	3
4	72	0	714430	1

df_selected 변수



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 군집별 고객 수를 확인하는 코드이다.

◆ 아래와 같이 6개 군집의 고객 수를 확인할 수 있음

```
cluster_counts = df['군집'].value_counts()  
print(cluster_counts)
```

군집

1	98
0	92
3	84
4	82
5	80
2	64

Name: count, dtype: int64



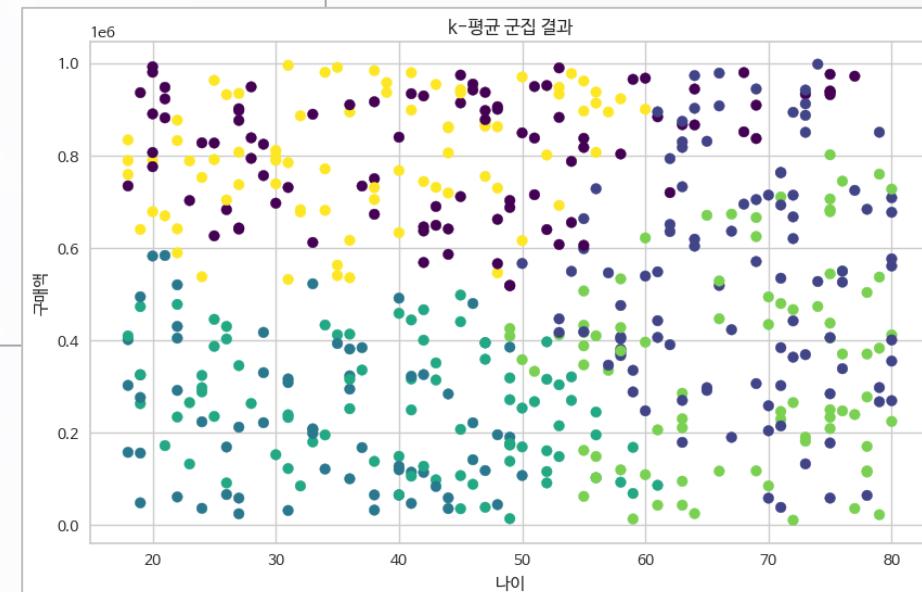
02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 군집별 시각화를 구현하는 코드이다.

- ◆ 아래 그림은 “나이”, “성별”, “구매액” 속성, K=6 값으로 K-평균 군집 결과 “나이” 와 “구매액” 군집별 시각화 결과임

— 시각화 결과에서 6개 군집으로 잘 분류되지 않는 것을 볼 수 있음

```
plt.figure(figsize=(14, 7))
plt.scatter(df['나이'], df['구매액'], c=df['군집'],
cmap='viridis', marker='o')
plt.title('Clusters of Customers')
plt.xlabel('나이')
plt.ylabel('구매액')
plt.colorbar(label='Cluster')
plt.show()
```





02 | 비지도 학습: K-평균 군집 분석

▲ 다음은 백화점 샘플 데이터 집합의 “나이”, “구매액” 속성으로 K-평균 군집 분석을 수행해 보자.

◆ 백화점 샘플 데이터 집합에서 “나이”, “구매액” 속성을 선택하여 변수 df_selected2에 할당함

```
# 필요한 속성 선택  
df_selected2 = df[['나이', '구매액']]  
  
df_selected2
```

	나이	구매액
0	49	518441
1	75	706071
2	19	473153
3	56	244905
4	72	714430
...
495	55	161926



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 표준화 함수로 데이터 정규화를 수행하는 코드이다.

◆ 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()  
df_scaled2 = scaler.fit_transform(df_selected2)  
df_scaled2
```

```
array( [[-0.02092837,  0.02289189],  
       [ 1.42624634,  0.66213514],  
       [-1.69074535, -0.13140139],  
       [ 0.36869559, -0.90902752],  
       [ 1.25926464,  0.69061371],  
       [-1.63508478,  0.94729901],
```



02 | 비지도 학습: K-평균 군집 분석

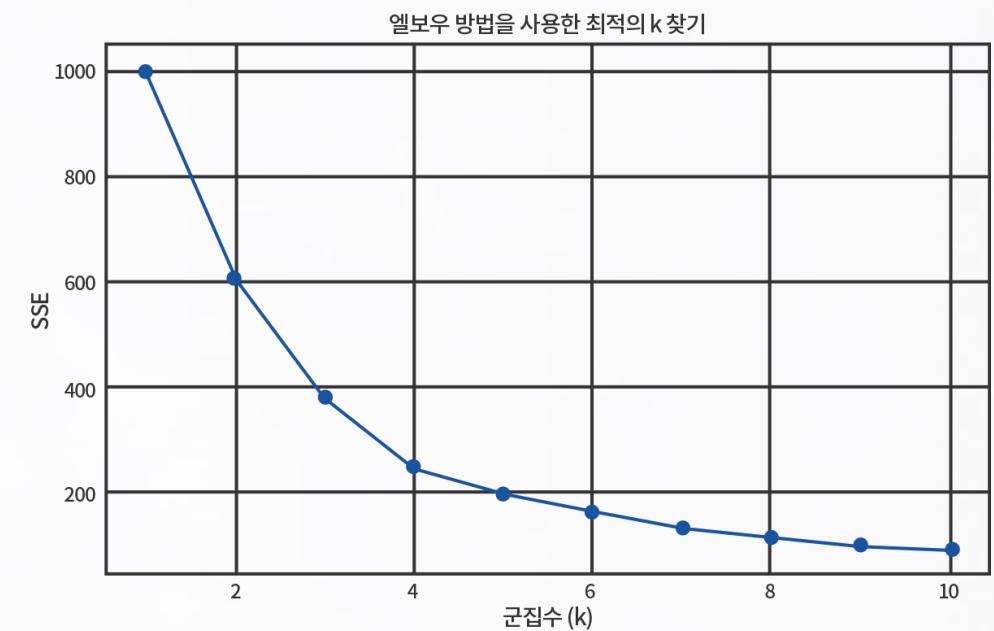
△ 다음은 엘보우(elbow) 방법을 이용해 최적의 K값을 찾는 코드이다.

◆ 최적의 K값을 찾기 위해 K=1~10까지 범위로 오차 제곱합(SSE)의 변화를 시각화함

— 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_scaled2)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse, marker='o')
plt.title('엘보우 방법을 사용한 최적의 k 찾기')
plt.xlabel('군집 수 (k)')
plt.ylabel('SSE')
plt.show()
```



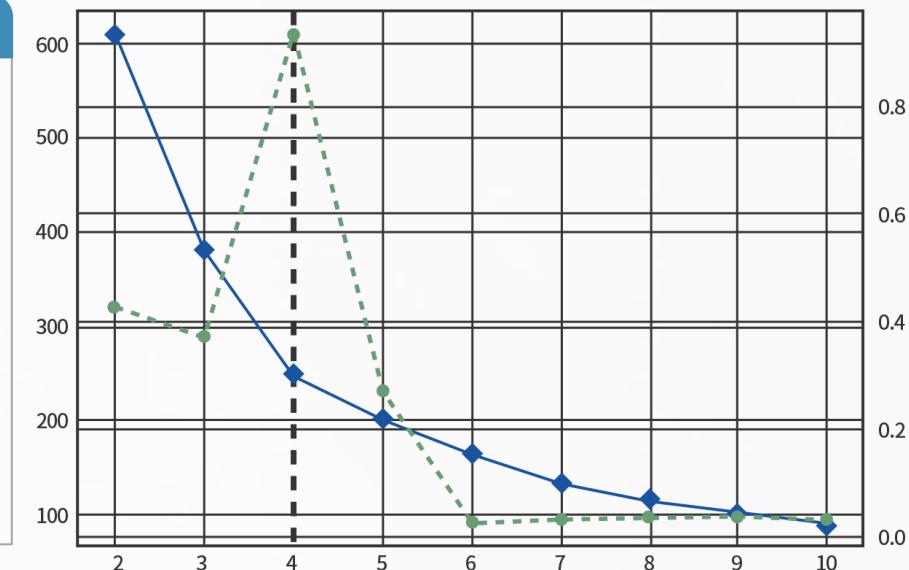


02 | 비지도 학습: K-평균 군집 분석

△ 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.

- ◆ 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
 - 검정색 점선의 위치를 보았을 때, 여기서는 K=4인 경우를 추천해주고 있음
 - 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer  
  
model = KMeans()  
visualizer = KElbowVisualizer(model, k=(2,11))  
visualizer.fit(df_scaled2)
```





02 | 비지도 학습: K-평균 군집 분석

- ❖ 다음은 최적의 K값을 4로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.
 - ◆ 아래와 같이 “나이”, “구매액” 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 4로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=4, random_state=42)  
kmeans.fit_predict(df_scaled2)  
  
# 각 군집의 중심값 확인  
centroids = kmeans.cluster_centers_  
centroids_df = pd.DataFrame(centroids,  
columns=df_selected2.columns)  
centroids_df
```

	나이	구매액
0	0.863456	0.877986
1	-0.769159	-0.897160
2	-0.876436	0.925922
3	0.888343	-0.822307



02 | 비지도 학습: K-평균 군집 분석

- ❖ 다음은 최적의 K값을 4로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.
- ◆ 아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 4로 가정하고 k-평균 군집 수행  
kmeans = KMeans(n_clusters=4, random_state=42)  
df['군집'] = kmeans.fit_predict(df_scaled2)  
df_selected2['군집'] = kmeans.fit_predict(df_scaled2)
```

	이름	나이	성별	직업	구매액	구매일자	거주지역	군집		나이	구매액	군집
0	김지후	49	여성	기타 행정 및 경영지원 관리자	518441	2023-09-07	송파구	1	0	49	518441	1
1	윤성현	75	여성	경영지원 관리자	706071	2024-02-23	관악구	0	1	75	706071	0
2	손성희	19	남성	임상 심리사 및 기타 치료사	473153	2024-03-15	서초구	1	2	19	473153	1
3	오정숙	56	남성	건설 및 광업 단순 종사원	244905	2023-07-06	종로구	3	3	56	244905	3
4	강채원	72	남성	물품 이동 장비 조작원	714430	2024-01-15	노원구	0	4	72	714430	0

df 변수

df_selected2 변수



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 군집별 고객 수를 확인하는 코드이다.

◆ 아래와 같이 4개 군집의 고객 수를 확인할 수 있음

```
cluster_counts = df['군집'].value_counts()  
print(cluster_counts)
```

군집

1 131
2 127
3 125
0 117

Name: count, dtype: int64



02 | 비지도 학습: K-평균 군집 분석

❖ 다음은 군집별 시각화를 구현하는 코드이다.

- ◆ 아래 그림은 “나이”, “구매액” 속성, K=4 값으로 K-평균 군집 결과
“나이” 와 “구매액” 군집별 시각화 결과임
 - 시각화 결과에서 **4개 군집**으로 **잘 분류된 것**을 볼 수 있음

```
plt.figure(figsize=(10, 6))
plt.scatter(df['나이'], df['구매액'], c=df['군집'], cmap='viridis')
plt.title('k-평균 군집 결과')
plt.xlabel('나이')
plt.ylabel('구매액')
plt.show()
```

