CAPSTON
Toy Project
비만-정상 클러스터링

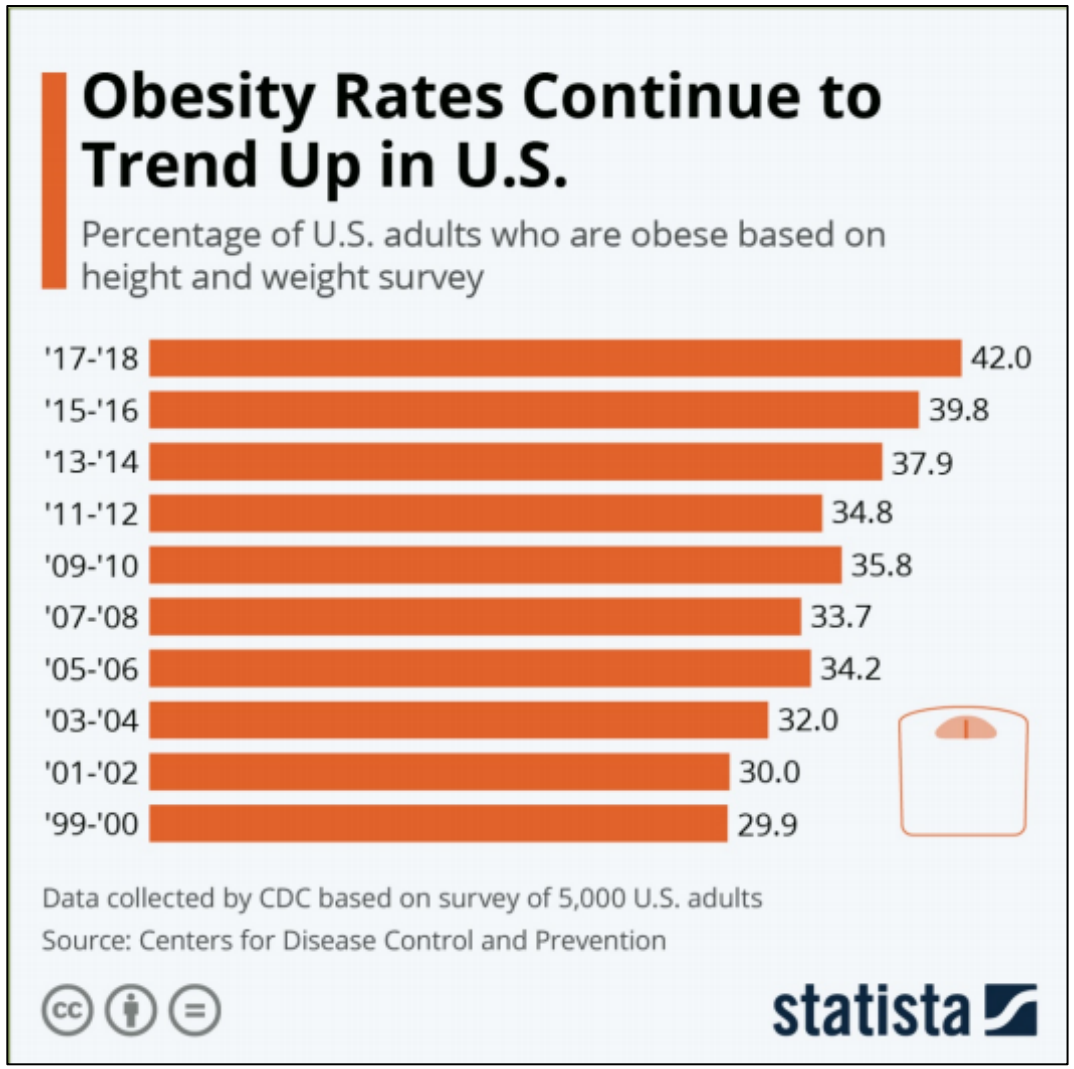
Ⅰ 프로젝트 개요

Ⅱ 프로젝트 수행절차

Ⅲ 프로젝트 사후평가

1. 프로젝트개요_기획의도

문제설정과 해결방안 제시



문제설정
✓ 과체중, 비만으로 인한 사망 증가

문제규명
✓ 해마다 높아지는 비만율

원인규명
✓ 고칼로리 음식 소비의 증가
✓ 줄어드는 신체활동

해결방안 → 1. 신체활동 증가
2. 체중감량 약물
3. 식이요법 적용

해결방안 제시 → 솔루션 개발

1. 프로젝트개요_내용

데이터 분석

데이터 전처리

다양한 모델 적용

적절한 모델선택

알고리즘 선택

평가지표 적용

결과 확인

결과값 예측

1. 프로젝트개요_환경 및 구조

언어

python

개발환경

jupyter

Visual Studio Code

데이터 분석 및 전처리

데이터 분석
- 상관관계 확인
- 특성 중요도 확인
- 비주요 특성 삭제

모델 적용

머신러닝 모델 적용

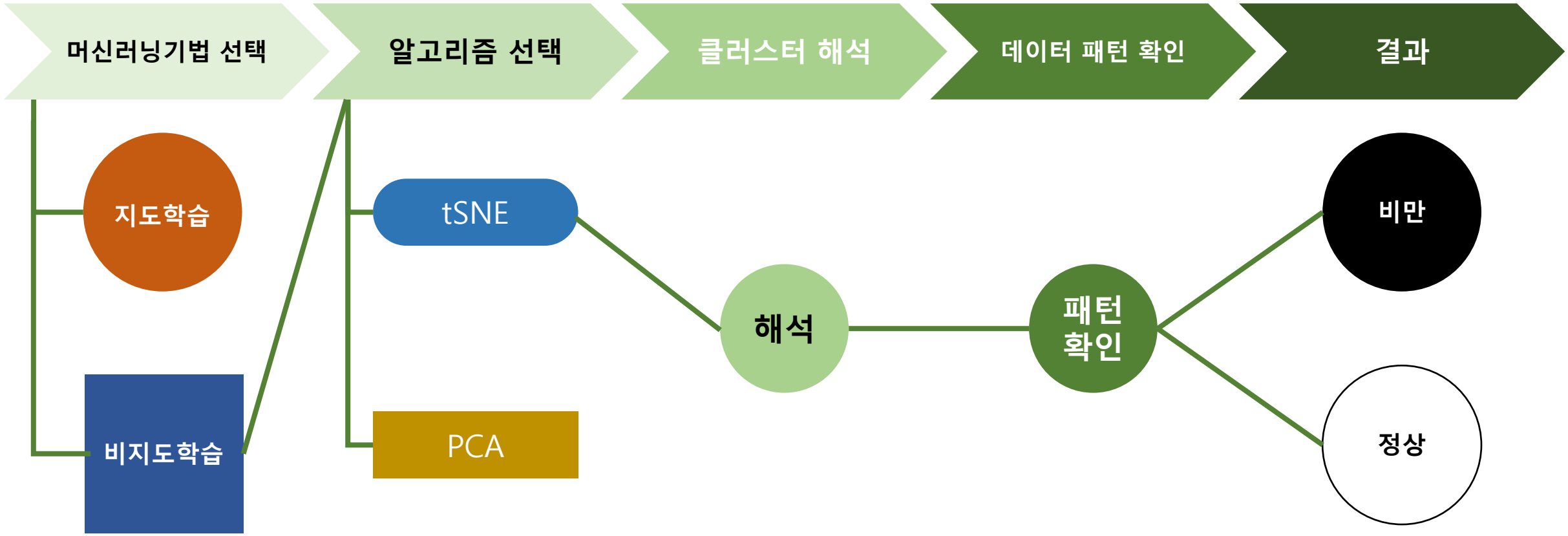
결과값 확인

결과값확인
- 라벨링 여부
- 평가지표 확인

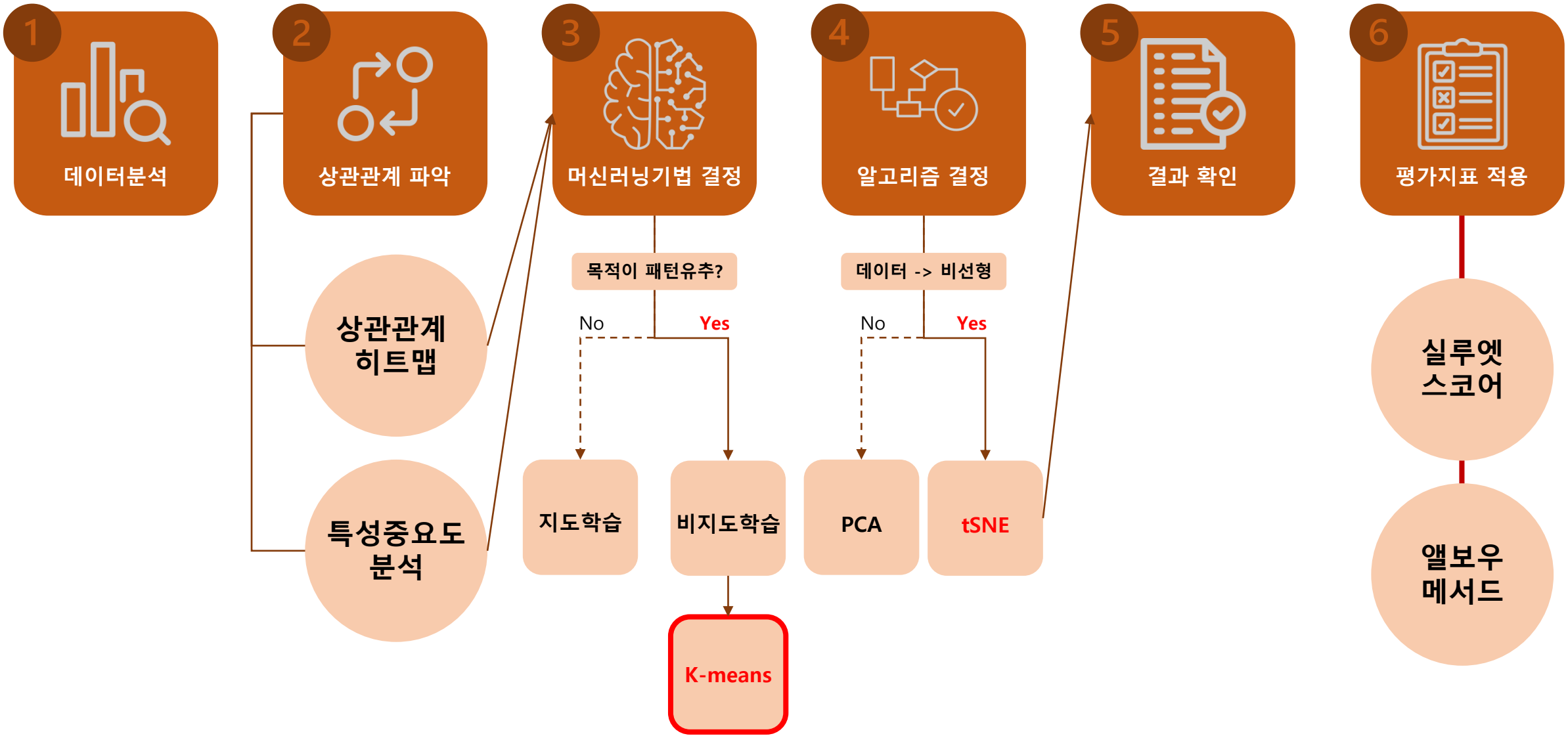
활용방안 논의

활용방안 논의
예측값으로 솔루션 개발
흥미로운 패턴발견 - 솔루션

1. 프로젝트개요_활용방안



2. 프로젝트 수행절차_Process



2. 프로젝트 수행절차_데이터분석

데이터분석

```
1 df
```

	activity	tBodyAcc.mean.X	tBodyAcc.mean.Y	tBodyAcc.mean.Z	tBodyAcc.std.X	tBodyAcc.std.Y	tBodyAcc.std.Z	tBodyAcc.mad.X	tBody
0	STANDING	0.279	-0.01960	-0.1100	-0.9970	-0.9670	-0.983	-0.997	
1	STANDING	0.277	-0.01270	-0.1030	-0.9950	-0.9730	-0.985	-0.996	
2	STANDING	0.277	-0.01470	-0.1070	-0.9990	-0.9910	-0.993	-0.999	
3	STANDING	0.298	0.02710	-0.0617	-0.9890	-0.8170	-0.902	-0.989	
4	STANDING	0.276	-0.01700	-0.1110	-0.9980	-0.9910	-0.998	-0.998	
...	...	...	...	...	...	...	...	...	
3604	WALKING_UPSTAIRS	0.357	-0.04460	-0.1300	-0.3140	-0.0556	-0.173	-0.386	
3605	WALKING_UPSTAIRS	0.344	0.00479	-0.1220	-0.3200	-0.0667	-0.182	-0.380	
3606	WALKING_UPSTAIRS	0.284	-0.00796	-0.1190	-0.3090	-0.0804	-0.211	-0.369	
3607	WALKING_UPSTAIRS	0.207	0.02460	-0.1040	-0.3650	-0.1690	-0.216	-0.449	
3608	WALKING_DOWNSTAIRS	0.393	-0.01780	-0.0902	-0.0963	-0.1740	-0.257	-0.153	

3609 행과 563개의 칼럼으로 이루어진 데이터

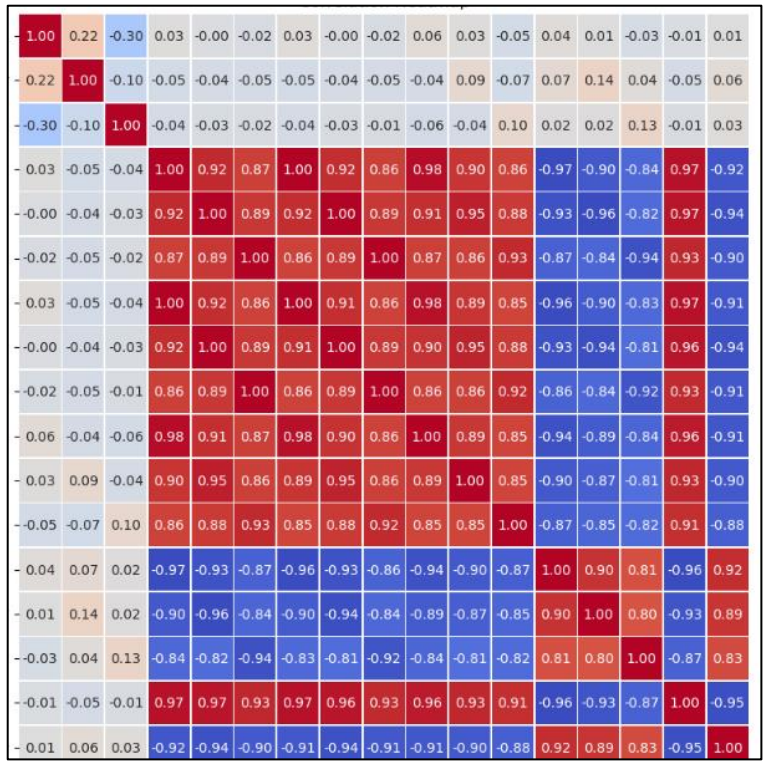
1. **'rn'칼럼 제외**
2. **'activity' 칼럼을 제외한 나머지컬럼은 전부 수치데이터**
3. X,y 라벨 각각 설정시 561차원의 그래프가 필요 → **불가능**

2. 프로젝트 수행절차_상관관계 파악

2

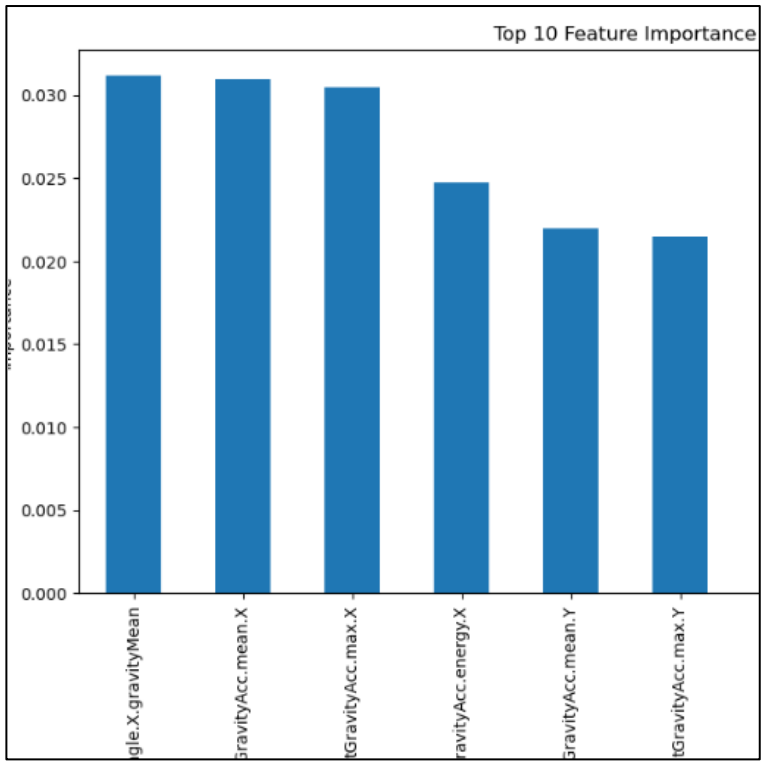
상관관계 파악

상관관계 히트맵



변수간 상관관계 분석
→ 563개의 데이터의 상관관계 有
→ 562차원의 그래프로 시각화 불가능

특성중요도



Y값 설정시(Activity)
X값의 특성중요도 파악
→ 최고값이 0.03이므로 지도학습 X

2. 프로젝트 수행절차_머신러닝기법 결정_K-means

3

머신러닝기법 결정

1. 사이킷_K-means 모델 적용

```
from sklearn.cluster import KMeans

# K-Means 모델 생성
kmeans = KMeans(n_clusters=2) # 클러스터 갯수 2
```

2. 스케일링 및 학습 (정규화)

```
# 데이터 준비 (예를 들면, 특성 선택 및 스케일링)
X = df.iloc[:, 2:]  # 데이터프레임의 필요한 열을 선택
# 데이터 스케일링 (옵션)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler() # 표준화 함수 객체 생성
X_scaled = scaler.fit_transform(X) # X를 정규화

# 모델 학습
kmeans.fit(X_scaled) # kmeans모델에 적용시키기
```

```
C:₩Users₩isfs0₩anaconda3₩lib₩site-packages₩sklearn₩cluster₩_
10 to 'auto' in 1.4. Set the value of `n_init` explicitly to
  super()._check_params_vs_input(X, default_n_init=10)
```

```
KMeans(n_clusters=2)
```

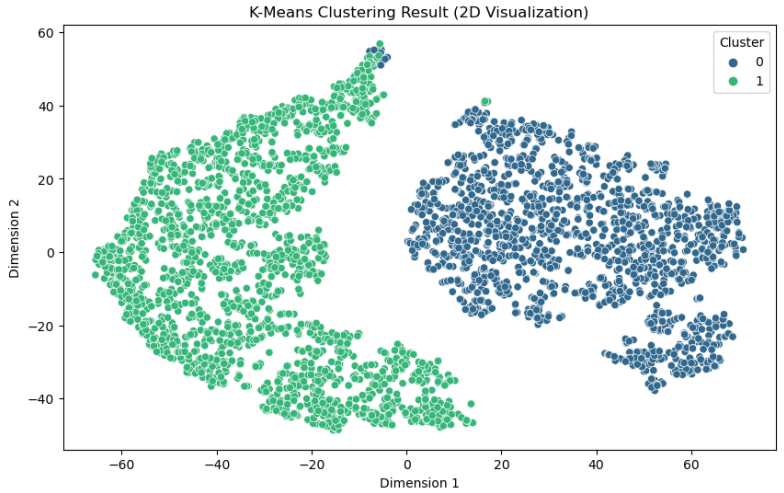
3. 정규화된 X값, K-means 예측값에 적용

```
# 클러스터 라벨을 정규화된 X값을 K-means예측모델에 적용
cluster_labels = kmeans.predict(X_scaled)
```

4. 클러스터 행렬에 라벨링

```
# 데이터의 클러스터 행렬에 라벨링
df['Cluster'] = cluster_labels
```

5. 시각화

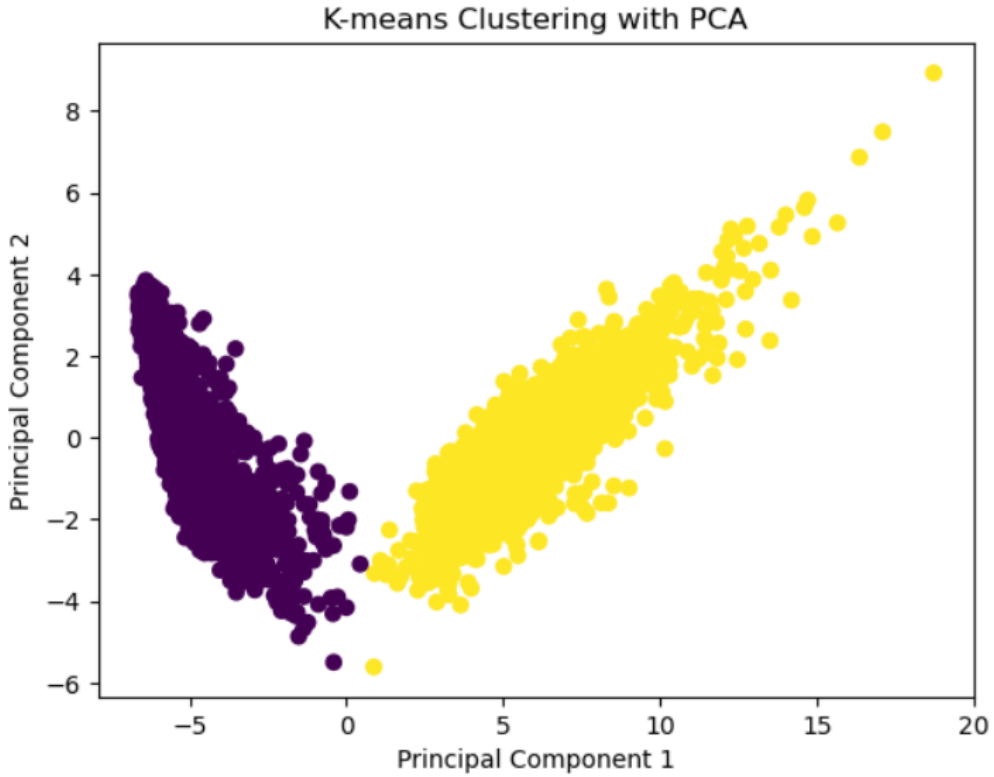


2. 프로젝트 수행절차_알고리즘 결정

tSNE 사용

- 563차원의 데이터를 2차원으로 축소
- 데이터의 소실량 적음
- 효과적으로 군집화

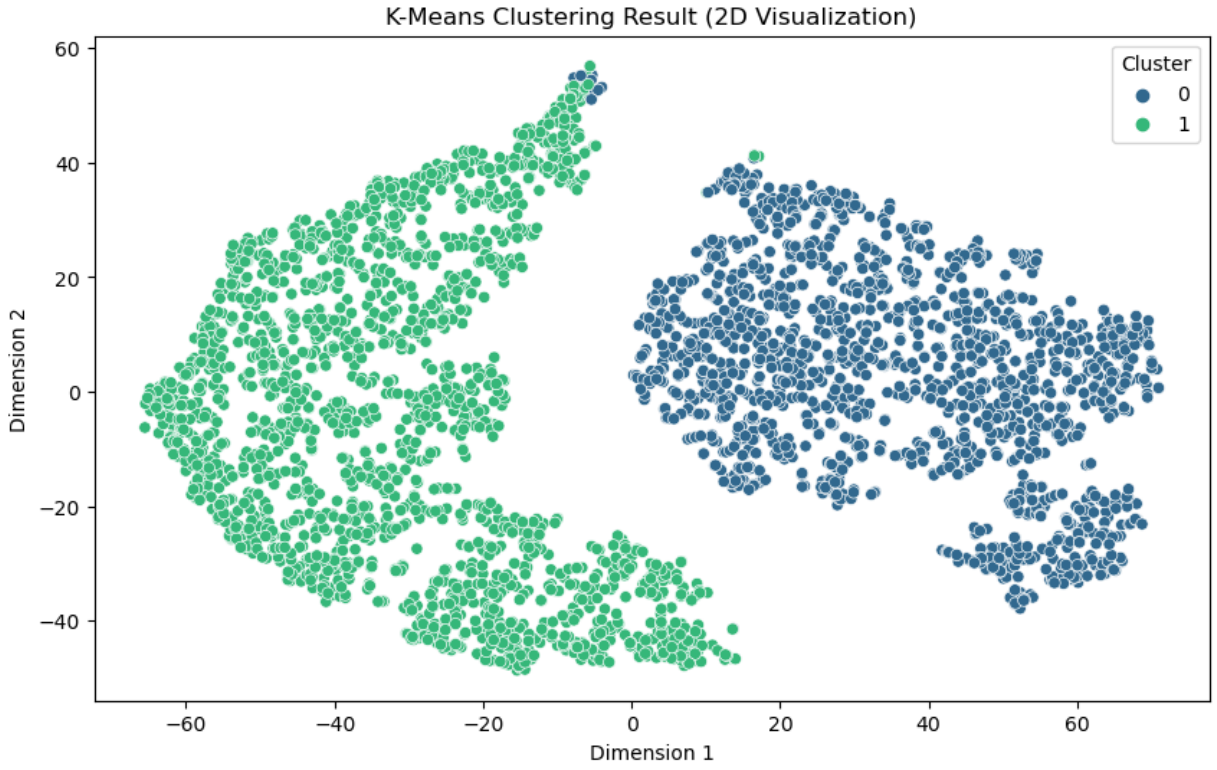
PCA 사용



- 563차원의 데이터를 2차원으로 축소
- 데이터의 소실량 많음
- 차원축소 과정에서 소실된 데이터 복구 불가

2. 프로젝트 수행절차_4) 알고리즘 / 5) 시각화

5. 시각화




```python
import seaborn as sns

# t-SNE를 사용하여 데이터를 2차원으로 시각화
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2, random_state=42)
X_tsne = tsne.fit_transform(X_scaled)

# 2D 산점도를 그리기 위한 데이터프레임 생성
df_tsne = pd.DataFrame(X_tsne, columns=['Dimension 1', 'Dimension 2'])
df_tsne['Cluster'] = cluster_labels

# Seaborn을 사용하여 클러스터별로 데이터 시각화
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Dimension 1', y='Dimension 2', hue='Cluster', data=df_tsne, palette='viridis')
plt.title('K-Means Clustering Result (2D Visualization)')
plt.show()
```

- Seaborn Library 사용(시각화)
- 클러스터 개수 : 2
- 알고리즘 : tSNE

2. 프로젝트 수행절차_결과 확인

Activity 값 라벨링 클러스터에 적용(코드)

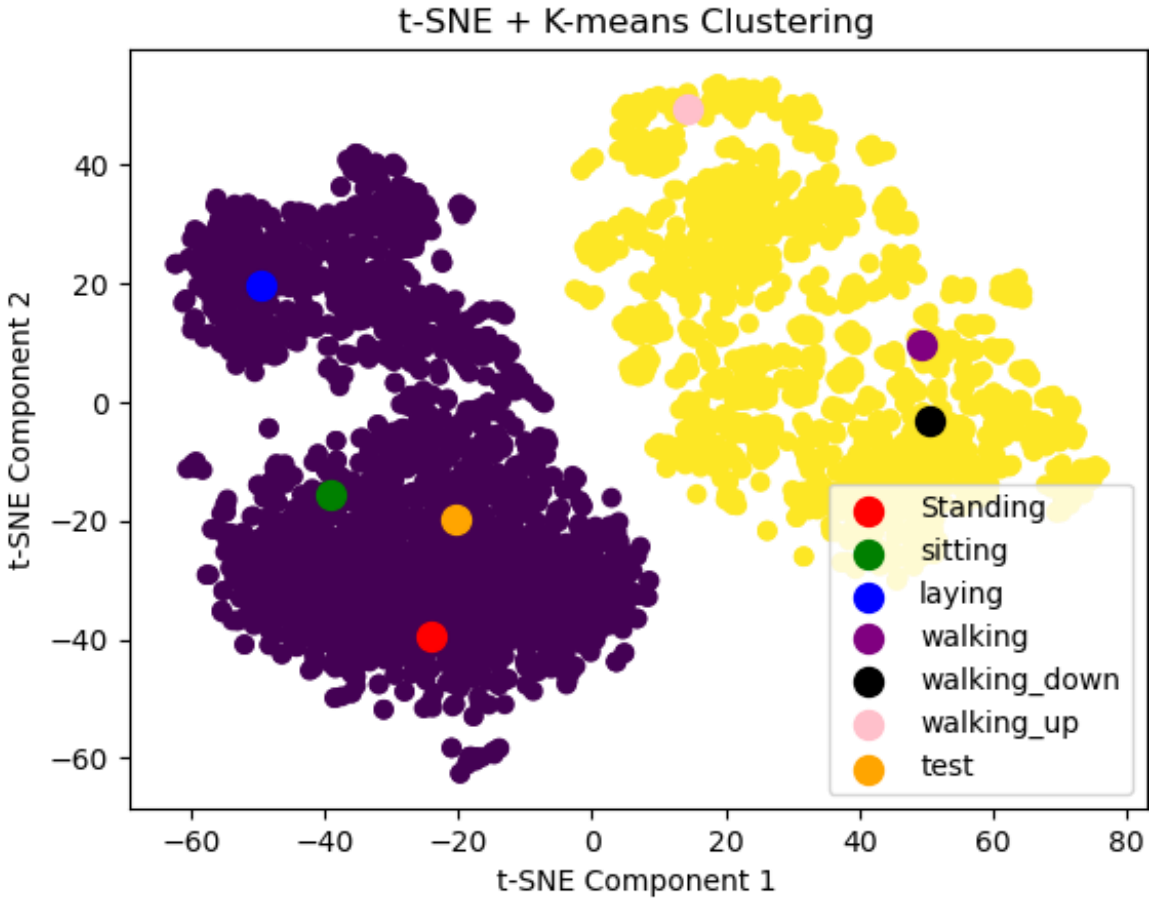
```
tsne_test = scaler.fit_transform(test)
tsne_test = tsne.fit_transform(tsne_test)


# 가정: selected_data_index는 선택한 데이터의 인덱스
selected_point = tsne_result[0]
selected_point_9 = tsne_result[586]
selected_point_17 = tsne_result[17]
selected_point_28 = tsne_result[28]
selected_point_43 = tsne_result[43]
selected_point_53 = tsne_result[53]
test_point = tsne_test[0]

# 시각화
plt.scatter(tsne_result[:, 0], tsne_result[:, 1], c=labels, cmap='viridis')
plt.scatter(selected_point[0], selected_point[1], color='red', s=100, label='Standing')
plt.scatter(selected_point_9[0], selected_point_9[1], color='green', s=100, label='sitting')
plt.scatter(selected_point_17[0], selected_point_17[1], color='blue', s=100, label='laying')
plt.scatter(selected_point_28[0], selected_point_28[1], color='purple', s=100, label='walking')
plt.scatter(selected_point_43[0], selected_point_43[1], color='black', s=100, label='walking_down')
plt.scatter(selected_point_53[0], selected_point_53[1], color='pink', s=100, label='walking_up')
plt.scatter(test_point[0], test_point[1], color='orange', s=100, label='test')
plt.title('t-SNE + K-means Clustering')
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.legend()
plt.show()

```

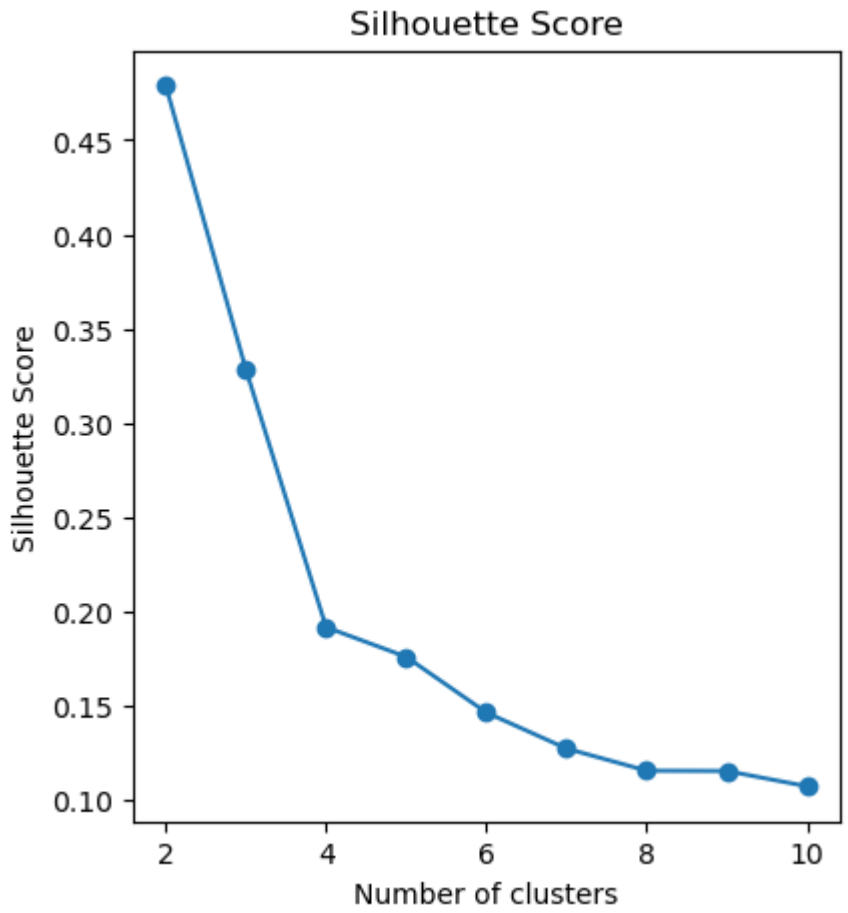
Activity 값 라벨링 클러스터에 적용(시각화)



라벨링 된(Activity)값이 정상적으로 클러스터에 포함됨을 확인

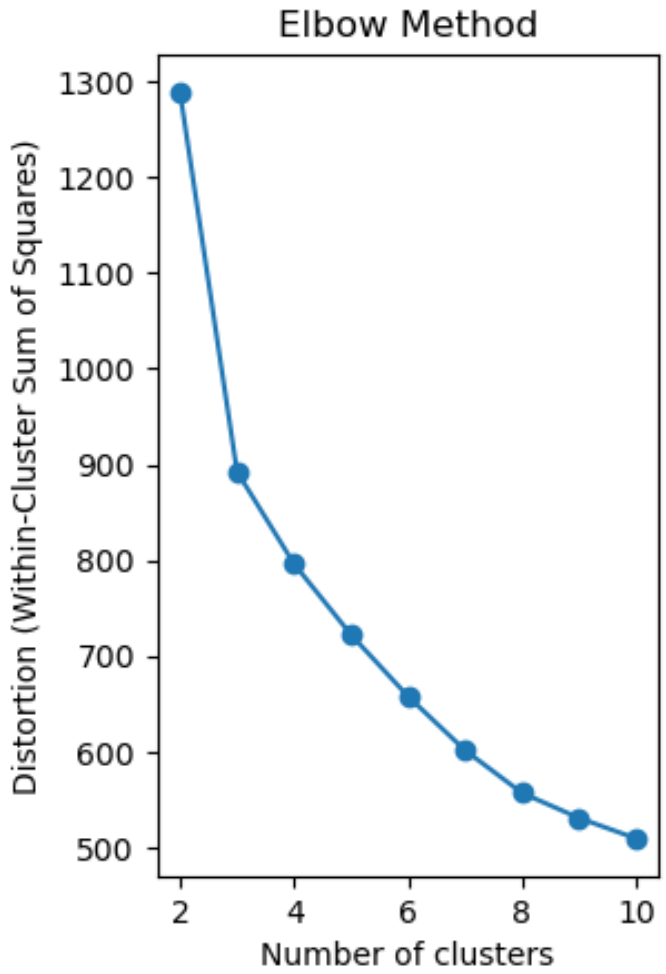
2. 프로젝트 수행절차_평가지표

실루엣 스코어 적용



실루엣스코어(0.5)는 클러스터값이 2일 때 높음

엘보우 메서드 적용



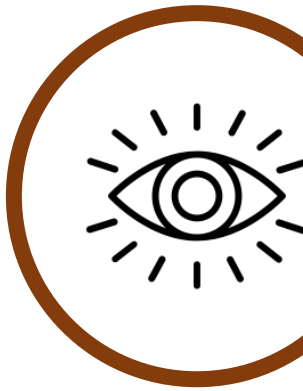
클러스터값이 2~4 사이일 때, 흥미로운 패턴을 발견

3. 사후평가

Good Point

분석력 향상
- 전체적인 데이터를 분석하는 과정에서, 학습 유형인 지도 학습과 비지도 학습 중, 데이터를 분석하는 데 있어 어떤 유형이 더 분석하기 유리한지 파악

비지도학습 심화탐구
- 많은 데이터들 간 유사성을 통해 숨겨진 패턴을 발견할 수 있는 비지도 학습 모델인 k-means 모델을 결정. 비지도학습의 알고리즘 공부 등 심층적으로 공부



소감

데이터 분석의 어려움
주제를 선정하고 데이터를 가져온 과정에서 데이터를 전처리 하기위해 분석하고 어떤 데이터인지에 대한 정의를 내리는 과정에서 어려움을 느낌

차원축소 학습의 어려움
데이터의 양과 특성이 너무 많았음. 처음 사용해본 차원축소 t-SNE를 이용해 데이터의 차원을 줄이고 군집화는 방법을 학습.

정규화의 필요성
데이터의 편차를 줄이는 과정이 필수라는 것을 알았다.