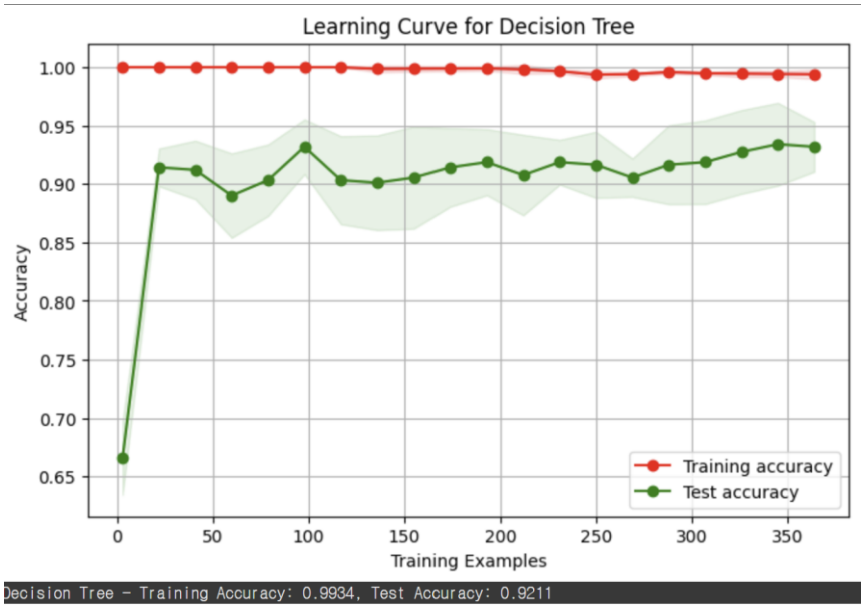
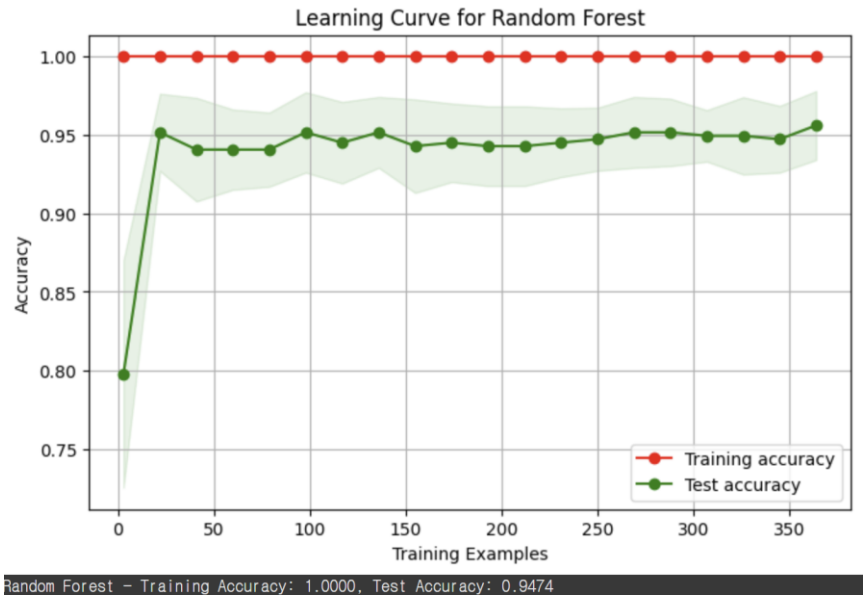
<report>

Ybigta 26th 김현운 ML과제 레포트

1.1)



Decision Tree - Training Accuracy: 0.9934, Test Accuracy: 0.9211



Random Forest - Training Accuracy: 1.0000, Test Accuracy: 0.9474

Random Forest 모델

훈련 정확도 (Training Accuracy): 100% (1.0000)

테스트 정확도 (Test Accuracy): 94.74% (0.9474)

훈련 데이터에 대해 완벽한 학습을 보이며, 과적합 가능성이 있지만 테스트 정확도가 높

은 수준으로 유지되고 있어, 일반화 성능이 우수하며 그래프에서 훈련 정확도와 테스트 정확도 간 차이가 적어 안정적이다.

Decision Tree 모델

훈련 정확도 (Training Accuracy): 99.34% (0.9934)

테스트 정확도 (Test Accuracy): 92.11% (0.9211)

훈련 정확도가 높은 반면, 테스트 정확도가 다소 낮아 일반화 성능이 비교적 떨어진다. 과적합의 가능성이 있으며, 테스트 정확도의 분산이 비교적 크므로 데이터에 대한 민감도가 높고 랜덤 포레스트보다 테스트 정확도가 낮아 일부 데이터에서 더 나쁜 성능을 보일 수 있음.

--그래프 분석

1. 훈련 정확도(빨간 선) 비교:
    - 두 모델 모두 훈련 정확도가 매우 높음(거의 100%).
    - 그러나 랜덤포레스트는 100% 정확도를 달성했으며, 이는 과적합의 신호일 수 있다.
2. 테스트 정확도(초록 선) 비교:
    - 랜덤포레스트의 테스트 정확도가 결정트리보다 더 높고 안정적인 경향을 보인다.
    - 랜덤포레스트의 경우, 테스트 정확도가 94.74%로 유지되며, 성능 편차가 적다.
    - 결정트리는 92.11%의 정확도를 기록했지만 변동 폭이 크고 불안정해 보인다.
3. 과적합 여부 분석:
    - 랜덤포레스트는 과적합이 존재하지만, 다수의 트리를 결합해 데이터의 노이즈를 줄이는 효과가 있어 테스트 정확도가 높다.
    - 결정트리는 훈련 데이터에 잘 적응하지만, 테스트 데이터에서 성능이 상대적으로 낮아진 것을 볼 수 있다.

1.2) 어느 모델이 더 나은가?

랜덤포레스트(Random Forest)가 더 나은 모델이다!!

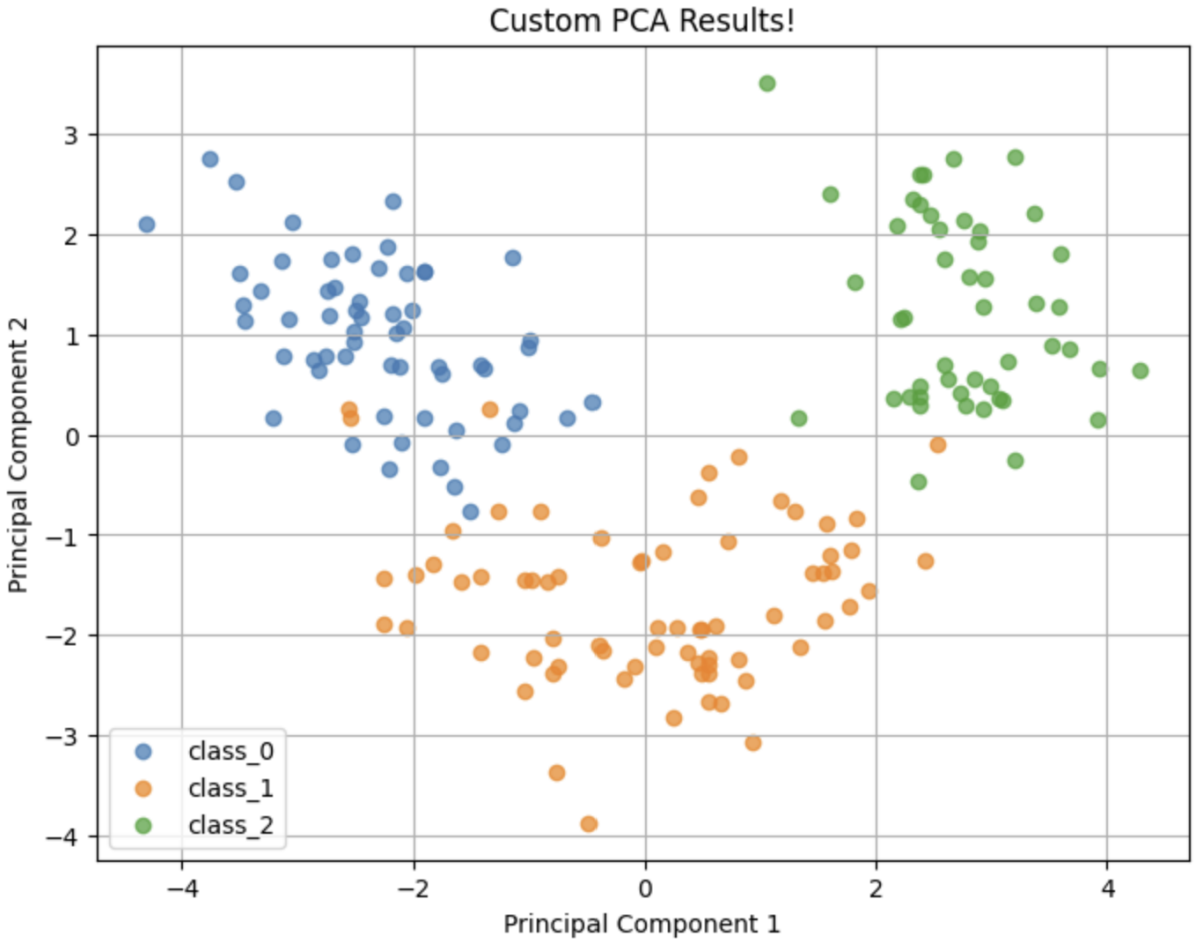
1: 더 높은 테스트 정확도 (94.74% vs 92.11%) → 더 좋은 일반화 성능.

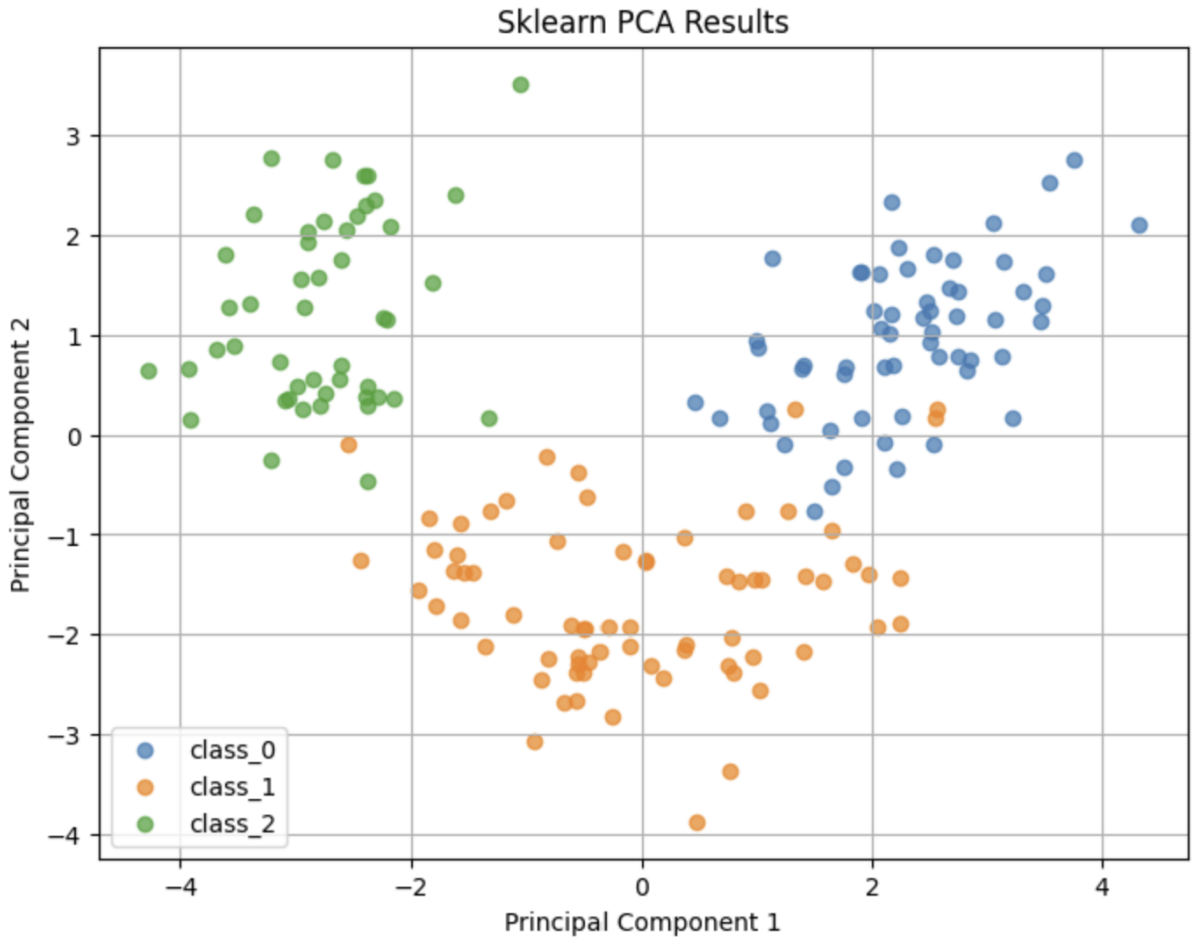
2: 결정 트리보다 분산이 적고 안정적인 성능을 유지한다.

3: 과적합을 완화하는 앙상블 기법 사용으로 신뢰성이 높다.

결론:
랜덤포레스트가 결정트리보다 더 나은 모델이며, 높은 테스트 정확도와 일반화 성능을 가지고 있다.

2.1)





위 plot들은 각각 제작한 커스터마이징 PCA와 SKlearn에서 기본적으로 제공하는 PCA의 2개의 주성분 분석 플랏이다. 먼저 standardizing 비슷한 평균 중심의 2차원 축을 제작하는 센터링을 진행하고, 공분산 행렬을 활용하여 eigenvalue와 eigenvector를 추출하여 차원축소에 활용한다.

2.2)

Pros and cons: 차원축소의 효과는 고차원 데이터를 저차원 데이터로 변환하여 계산 시간과 메모리를 줄이고, 시각화가 용이하도록 하며 모델성능을 향상시킬 수 있다. 과적합을 방지하는 과정의 일환이기 때문이다. 노이즈가 제거된다는 것은 장점으로 작용하나 정보 손실의 가능성도 존재한다. 일부의 주성분만을 유지한다면 원래 데이터의 중요 정보가 소실될 수 있다. 또한 원래 데이터로부터 새로운 차원으로 변형을 가한것이기 때문에 해석의 어려움이 있을 수 있고, 비선형 데이터는 처리가 어렵다, PCA는 선형관계에 기반한 처리법이기 때문이다.

Alternative method: "T-SNE"

t-distributed stochastic neighbor embedding의 약자로, 고차원 데이터를 저차원 공간에 mapping하면서 데이터의 국소적 구조를 보존하는 차원축소 법이다. 클러스터링과 궤를 같이 한다. PCA는 선형성이 담보되어야 용이했다면 이는 비선형적 데이터구조를 포착하여 복잡한 패턴을 효과적으로 시각화하고, 지역적 군집성과 유사성을 보존하기에 군집탐색에 효과적이다. 그러나 비선형적 처리가 가능하기에 계산비용이 높고, 새로운 데이터에 대해 적응이 힘들다는 특징이 있다고 한다.

3.1)

1. 마진(Margin)이란?

서포트 벡터 머신(SVM)에서 **마진(Margin)**은 데이터를 분류하는 초평면(Decision Boundary)과 가장 가까운 데이터 포인트(서포트 벡터) 사이의 거리다.

- SVM의 목표는 **마진을 최대화**하여 데이터의 일반화 성능을 극대화하는 것이다.
- 마진이 넓을수록 모델이 새로운 데이터를 더 잘 일반화할 수 있다.
- 두 개의 마진 경계 사이에 데이터가 없어야 한다.

하드 마진과 소프트 마진은 이러한 초평면과 데이터 간에 제약조건(constraint)를 어떻게 부과하는지에 따라 구분 가능하다고 교육세션때 배운 바 있다. SV와 하이퍼플레인 간의 거리가 일부 데이터와 하이퍼플레인 간의 거리보다 클 수도 있는 것이 소프트 마진 SVM이다.

---

2. 하드 마진 SVM (Hard Margin SVM)

하드 마진 SVM은 **마진을 최대화하면서 모든 데이터 포인트가 올바르게 분류**되는 경우를 의미한다. 즉, **오차 허용 없이** 데이터를 완벽하게 분리해야 한다.

특징:

- 선형적으로 완벽하게 분리 가능한 경우에 적용.
- 오차를 허용하지 않으므로 노이즈가 없는 데이터에 적합.

장점:

1. **완벽한 분류 가능** – 선형적으로 분리 가능한 데이터에서는 최적의 결정 경계를 제공.
2. **단순한 모델** – 수학적으로 해석이 명확하고 계산량이 적음.

단점:

1. **과적합(Overfitting) 위험** – 훈련 데이터에 완벽하게 맞추려다 보니 일반화 성능이 낮아질 수 있음.
2. **노이즈 민감성** – 이상치(outliers)나 노이즈가 있으면 모델 성능이 크게 저하됨.

---

3. 소프트 마진 SVM (Soft Margin SVM)

소프트 마진 SVM은 **약간의 오차를 허용하면서 마진을 최대화**하는 방식이다. 즉, 일부 데이터가 잘못 분류될 수 있도록 허용하면서도 마진을 넓히는 것을 목표로 한다.

특징:

- 선형적으로 완벽하게 분리되지 않는 데이터셋에서도 적용 가능.
- 일부 오차를 허용하여 일반화 성능을 높임.

장점:

1. **노이즈에 강함** – 일부 오차를 허용하므로 이상치(outliers)로 인한 영향이 적음.
2. **비선형 데이터 처리 가능** – 데이터를 선형적으로 분리할 수 없는 경우에도 좋은 성능을 보임.
3. **일반화 성능 향상** – 적절한 오차 허용으로 테스트 데이터에 대한 성능이 좋아짐.

단점:

1. **모델 복잡도 증가** – 최적의 마진과 오차 허용 정도를 조정해야 하므로 튜닝이 필요.
2. **과소적합(Underfitting) 위험** – 너무 많은 오차를 허용하면 성능이 저하될 수 있음.

---

4. 하드 마진 vs 소프트 마진

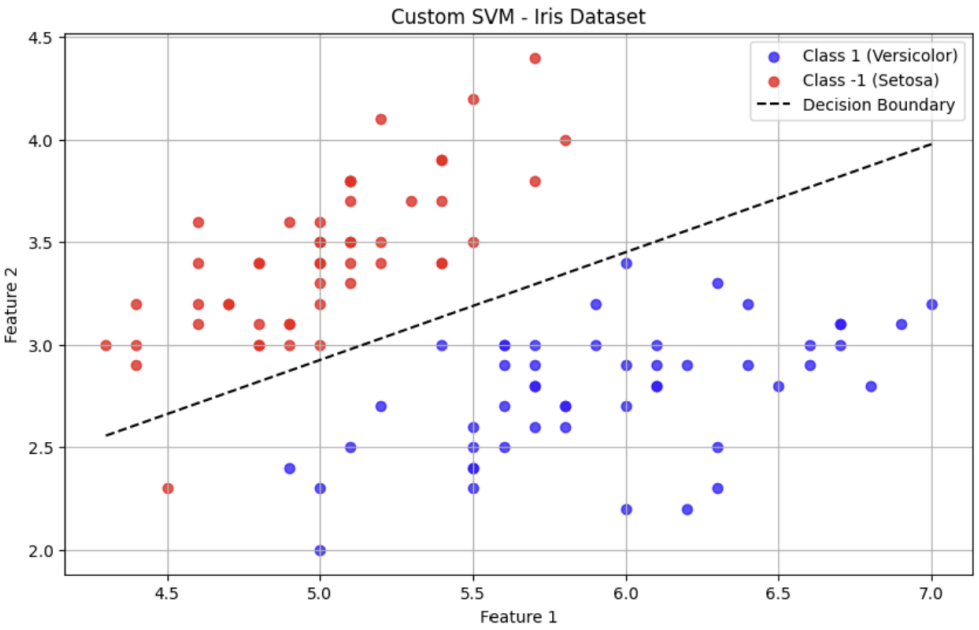
특징	하드 마진 SVM	소프트 마진 SVM
데이터 분리 여부	선형적으로 완벽히 분리 가능할 때	일부 오차를 허용하여 분리
오차 허용	없음	있음
노이즈 민감도	민감함	덜 민감함
일반화 성능	낮을 수 있음 (과적합 위험)	높음 (과적합 방지)
사용 시기	이상치 없는 선형 분리 가능	이상치 존재 및 비선형 데이터

---

5. 결론 및 적용 시나리오

- **하드 마진 SVM은** 데이터가 선형적으로 분리 가능하고 노이즈가 없는 경우 적합.
- **소프트 마진 SVM은** 실제 데이터를 다룰 때, 노이즈나 겹치는 데이터가 존재하는 경우 일반적으로 더 적합.
- 대부분의 현실 데이터는 완벽하게 분리되지 않기 때문에 **소프트 마진 SVM**이 더 많이 사용됨.

3.2)



Loss function을 동일하게 사용하여 구현했기에 제시된 그래프 플랏과 형태가 동일하다.