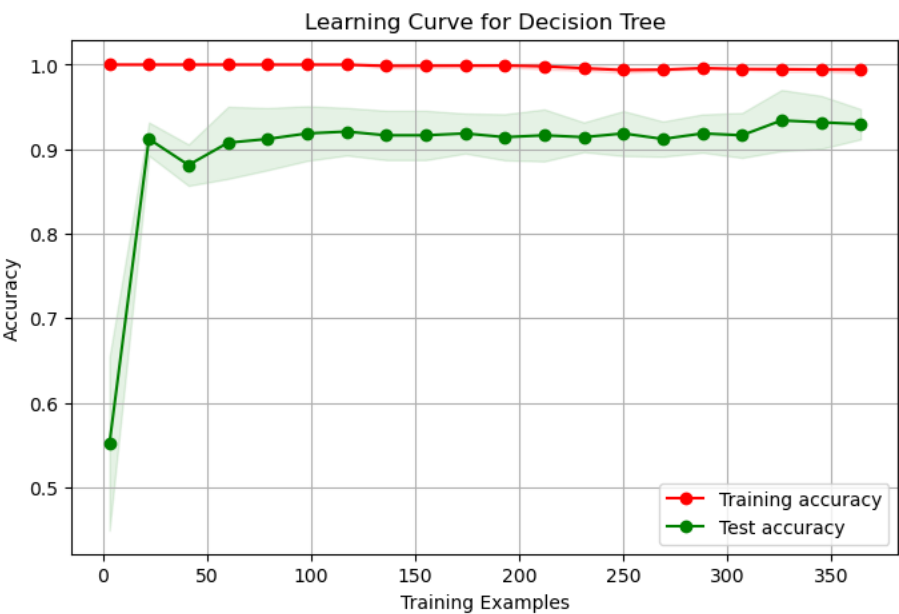


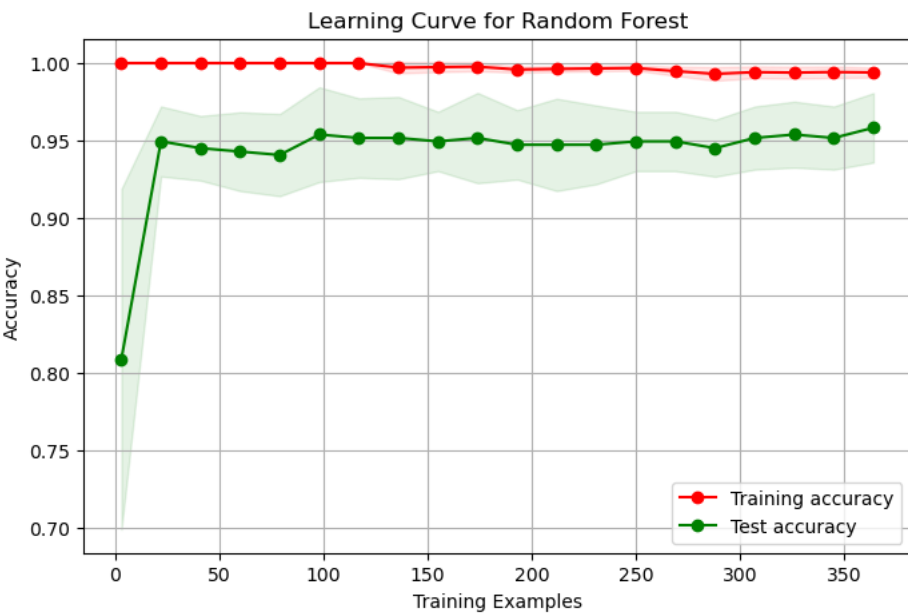
ML_Report

2025/01/28

2024148030 김정인



Decision Tree - Training Accuracy: 0.9934, Test Accuracy: 0.9123



Random Forest - Training Accuracy: 0.9934, Test Accuracy: 0.9561

◆ Decision Tree vs. Random Forest: 어느 모델이 더 좋은가?

주어진 정보에서 **Decision Tree**와 **Random Forest**를 비교하여 더 나은 모델을 판단하자.

1. 정확도 비교

테스트 데이터에서의 정확도를 비교하면:

모델	학습 정확도 (Training Accuracy)	테스트 정확도 (Test Accuracy)
Decision Tree	99.34%	91.23%
Random Forest	99.34%	95.61%

- **Random Forest가 테스트 정확도가 더 높음 (95.61% vs. 91.23%)**
→ 즉, 새로운 데이터에서도 더 좋은 성능을 보인다.
- 학습 정확도는 두 모델이 동일하지만, **Random Forest는 일반화 성능이 더 좋음.**

2. Overfitting(과적합) 문제

- Decision Tree는 개별 트리 하나만 사용하기 때문에 **과적합 가능성이 높음.**
- Random Forest는 여러 개의 트리를 결합하여 **과적합을 방지하는** 효과가 있음.

→ **Random Forest가 과적합을 방지하며 일반화 성능이 뛰어남.**

3. 모델의 안정성과 일반화 성능

모델	안정성	일반화 성능
Decision Tree	낮음 (데이터 변화에 민감)	중간
Random Forest	높음 (여러 트리 평균)	높음

- **Decision Tree**는 하나의 트리만 학습하기 때문에 **데이터가 조금만 변해도 예측이 달라질 가능성이 높음.**
- **Random Forest**는 여러 트리를 조합하여, 하나의 트리가 잘못된 예측을 하더라도 다른 트리들이 보정하는 역할을 함.
- Random Forest가 **데이터 변화에 덜 민감하고 더 안정적인 예측 가능.**

⚙ 4. 계산 복잡도 및 속도

모델	학습 속도	예측 속도
Decision Tree	빠름	매우 빠름
Random Forest	느림 (여러 트리 학습)	빠름

- **Decision Tree**는 학습이 빠르지만, 성능이 떨어질 수 있음.
- **Random Forest**는 여러 개의 트리를 학습해야 하므로 학습 시간이 더 오래 걸리지만, **예측 속도는 빠름**.
- 속도가 중요하다면 Decision Tree가 유리하지만, **성능을 고려하면 Random Forest가 더 좋음**.

5. 특징 중요도(Feature Importance)

- **Random Forest**는 여러 트리에서 중요 특징을 분석하여, 더 신뢰할 수 있는 **Feature Importance**를 제공.
- **Decision Tree**는 특정 데이터에 최적화되므로, 중요한 변수를 과대평가하거나 과소평가할 가능성이 있음.
- 특징 중요도를 분석할 때 **Random Forest**가 더 신뢰할 수 있음.

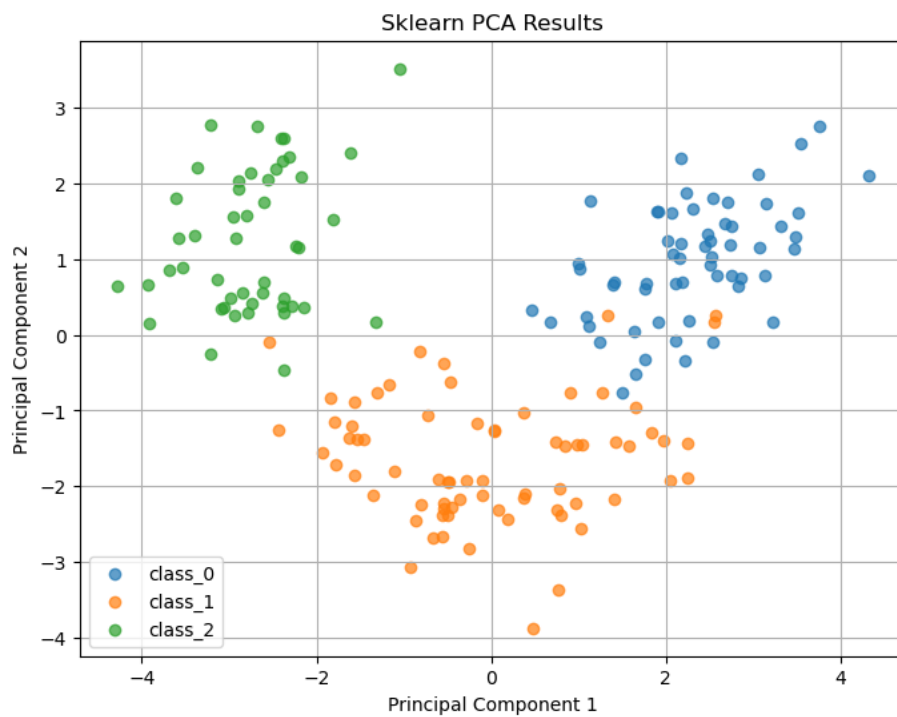
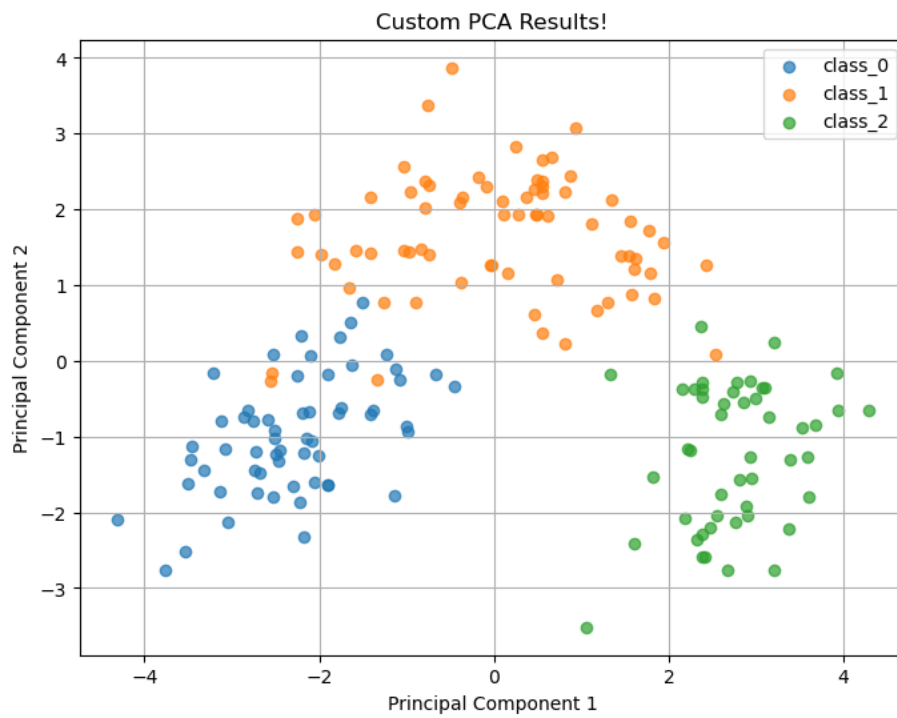
최종 결론: Random Forest vs. Decision Tree

모델	정확도	과적합 방지	안정성	속도	해석 가능성
Decision Tree	낮음 (91.23%)	과적합 위험 높음	낮음	빠름	높음
Random Forest	높음 (95.61%)	과적합 방지	높음	느림 (학습) / 빠름 (예측)	낮음

Random Forest가 더 좋은 선택

- 일반화 성능(테스트 정확도)이 더 높음.

- 과적합 위험이 적고, 안정적인 모델.
- 데이터가 조금 변해도 예측이 크게 바뀌지 않음.
- 특징 중요도를 신뢰할 수 있음.



◆ Principal Component Analysis (PCA)의 장점과 단점

PCA(주성분 분석)는 차원 축소(dimensionality reduction) 기법으로, 데이터의 분산을 최대화하는 방향으로 새로운 좌표축(주성분, Principal Components)을 찾아 데이터를 변환하는 방법이다. 하지만, 모든 문제에 적합한 것은 아니므로 장점과 단점을 잘 이해하는 것이 중요하다.

✅ PCA의 장점

1. 차원 축소(Dimensionality Reduction)

- 데이터를 더 적은 차원으로 변환하여 계산 속도를 향상시키고, 메모리 사용량을 줄일 수 있음.
- 예: 100차원의 데이터를 2~3차원으로 줄여 시각화할 수 있음.

2. 노이즈 제거(Noise Reduction)

- 데이터의 주요 변동성을 보존하면서, 잡음(노이즈)이나 중요하지 않은 특성을 제거할 수 있음.
- 특히, 고차원 데이터에서 성능 향상에 도움을 줄 수 있음.

3. 다중공선성(Multicollinearity) 문제 해결

- 독립변수들 간의 강한 상관관계를 줄여서 회귀 분석이나 머신러닝 모델의 성능을 개선할 수 있음.

4. 시각화 가능

- 데이터의 주요 패턴을 2D 또는 3D로 변환하여 시각화할 수 있음.

5. 차원이 높은 데이터를 다룰 때 유용

- 예를 들어, 이미지 데이터(픽셀 수가 매우 많음)나 유전자 데이터와 같이 차원이 높은 데이터를 다룰 때 효과적임.

❌ PCA의 단점

1. 해석이 어려움(Interpretability)

- 변환된 데이터(주성분, Principal Components)는 원래 변수와 직접적인 의미가 연결되지 않음.

- 예를 들어, 원래 데이터의 특정 특성이 어떤 주성분에 어떤 영향을 미치는지 해석하기 어려울 수 있음.

2. 비선형 데이터에는 적합하지 않음

- PCA는 데이터가 선형적으로 분리 가능한 경우 효과적이지만, 복잡한 비선형 데이터에서는 성능이 낮을 수 있음.

3. 정보 손실 가능성

- 주성분을 줄이는 과정에서 일부 중요한 정보가 손실될 수 있음.
- 너무 적은 차원으로 축소하면 데이터의 본질적인 특성을 잃을 위험이 있음.



4. 데이터 정규화 필요

- PCA를 적용하기 전에, 데이터의 스케일(표준화 또는 정규화)을 맞추지 않으면 결과가 왜곡될 수 있음.

PCA 이외의 차원 축소 기법

PCA 외에도 다양한 차원 축소 방법이 존재하며, 특히 **비선형 데이터**나 **고차원 데이터**를 다룰 때 더 적합한 방법이 있음.

1. t-SNE (t-Distributed Stochastic Neighbor Embedding)

- 비선형 차원 축소 기법으로, 특히 고차원 데이터를 2D 또는 3D로 시각화할 때 자주 사용됨.
- 데이터의 지역적인 구조(local structure)를 보존하는 특징이 있음.
- 주로 이미지 데이터, 유전자 데이터, 자연어 처리(NLP) 등에 사용됨.
-  **장점:** 데이터의 군집(cluster)을 잘 나타낼 수 있음.
-  **단점:** 학습 시간이 길고, 새로운 데이터를 추가하는 것이 어려움.

◆ Soft Margin SVM vs. Hard Margin SVM 차이점

Support Vector Machine(SVM)은 분류 문제에서 데이터를 두 개의 클래스로 나누는 초평면(hyperplane)을 찾는 기법이다.

이때, **Hard Margin SVM**과 **Soft Margin SVM**은 마진(margin)을 설정하는 방식이 다르다.

1. Hard Margin SVM

- 모든 데이터가 마진(margin) 바깥쪽에 있어야 한다.
- 즉, 완벽하게 선형적으로 분리(linearly separable)되는 데이터에만 적용 가능하다.
- 서포트 벡터들이 결정 경계(hyperplane)와 마진 사이에 위치해야 하며, 오차(허용 오류, slack variable)를 허용하지 않음.

Hard Margin SVM의 장점

1. 완벽하게 선형적으로 분리되는 데이터에서 강력한 결정 경계를 형성한다.
2. 오류를 허용하지 않으므로 높은 신뢰도를 제공할 수 있다.
3. 과적합(overfitting) 가능성이 낮다 (단, 데이터가 깨끗한 경우).

Hard Margin SVM의 단점

1. 노이즈가 있는 데이터나 선형적으로 분리되지 않는 데이터에 취약하다.
 2. 오버피팅(overfitting)이 발생할 가능성이 있음.
 3. 데이터가 완벽히 분리되지 않으면 해결이 불가능함.
-

2. Soft Margin SVM

- 현실적인 데이터는 완벽하게 선형적으로 분리되지 않는 경우가 많기 때문에, **약간의 오차(허용 오류, slack variable)를 허용하는 방식**.
- Hard Margin SVM과 달리, 일부 데이터 포인트가 마진을 넘어서거나 결정 경계를 침범하는 것을 허용.
- **C-regularization** 파라미터를 사용하여 오차를 얼마나 허용할지를 조정할 수 있음.

✅ Soft Margin SVM의 장점

1. 노이즈가 있는 데이터에서도 잘 작동함.
2. 선형적으로 분리되지 않는 데이터에도 적용 가능.
3. 일반화(generalization) 성능이 우수하여 새로운 데이터에 대한 예측이 더 안정적.

❌ Soft Margin SVM의 단점

1. C 값 조정이 필요하여 최적의 하이퍼파라미터 설정이 어려울 수 있음.
2. 과적합(overfitting) 위험이 있음 (C 값을 너무 작게 설정하면 일반화 성능이 떨어질 수도 있음).
3. 계산 비용이 더 큼 (Hard Margin보다 복잡한 계산이 필요).

📌 3. Soft Margin vs. Hard Margin 정리

비교 항목	Hard Margin SVM	Soft Margin SVM
허용 오차 (Slack Variable)	허용하지 않음 (오류 0)	허용 (오류 가능)
적용 가능 데이터	선형적으로 완벽히 분리 가능한 경우	노이즈가 있거나 선형적으로 분리되지 않는 경우
과적합(Overfitting)	가능성이 있음	일반적으로 방지됨
일반화 성능	데이터가 깨끗할 경우 뛰어남	다양한 데이터에 적용 가능
계산 비용	낮음	높음 (최적화 필요)

