

앙상블 고급 모형

1. 앙상블 모델 구조

1. 앙상블 모델 구조

앙상블 모델은 여러 개의 개별 모델을 결합하여 더 나은 성능을 얻는 방법론이다. 개별 모델들이 결합하여 최종 결정을 내리기 때문에 단일 모델보다 예측 성능이 더 뛰어난 경우가 많다.

앙상블 모델은 은 결정트리 기반 알고리즘을 결합하여 구현

앙상블 모델의 개념

- 앙상블 모델은 여러 개의 모델을 결합하여 단일 모델보다 높은 예측 성능을 달성하는 방법론이다.
- 개별 모델들의 예측 결과를 종합하여 최종 결정을 내린다.
- 단일 모델보다 예측 성능이 뛰어나고, 과적합을 방지할 수 있다.

1. 앙상블 모델 구조

앙상블 모델 종류 요약

1. **Voting** : 서로 다른 알고리즘 가진 분류기를 결합,
사이킷런은 VotingClassifier 클래스 라이브러리를 제공함
 - (1) **하드보팅(Hard Voting)** :
분류기들이 예측한 결과 값을 다수결로 결정한다
 - (2) **소프트 보팅(Soft Voting)** :
각 분류기들이 예측 값을 확률로 구하면 이를 평균 내어 확률이 가장 높은 값을 결과 값으로 결정

1. 앙상블 모델 구조

2. Bagging :

같은 유형의 알고리즘을 결합, 데이터 샘플링시 서로 다르게 가져가면서 학습, RandomForest 가 대표적이다,
Bootstrapping Aggregation 줄임말 (Bootstrapping : 여러 개의 데이터 세트를 중첩되게 분리하는 분할 방식)

3. Boosting :

여러 개의 분류기가 순차적으로 학습하면서 가중치를 부스팅한다.
XGBoost (캐글 대회 상위 석권),
LightGBM (마이크로 소프트사 제공)

1. 앙상블 모델 구조

배깅(Bagging)

- 배깅은 여러 모델을 독립적으로 학습시킨 후, 그 결과를 평균 내거나 다수결로 결정하는 방식이다.
- 각 모델은 데이터의 일부 샘플과 특성의 일부를 무작위로 선택하여 학습한다.
- 이를 통해 모델의 분산을 줄이고, 과적합을 방지한다.
- 대표적인 배깅 알고리즘은 RandomForest이다

랜덤포레스트(RandomForest)

- 의사 결정 트리 기반(Decision Tree) 기반 분류 알고리즘
- 앙상블(Ensemble), 같은 결정트리를 여러 개 사용, 비교적 빠른 수행
- 현재의 랜덤 포레스트의 개념은 레오 브레이먼(Leo Breimen)의 논문에서 만들어짐, 이 논문은 랜덤 노드 최적화(Randomized Node Optimization, RNO)와 배깅(bagging)을 결합한 방법과 같은 CART(Classification And Regression Tree)를 사용해 상관관계가 없는 트리들로 포레스트를 구성하는 방법을 제시했다

1. 앙상블 모델 구조

RandomForest 모델 소개

RandomForest는 배깅(Bagging) 기법을 활용한 앙상블 모델이다. 여러 개의 결정 트리(Decision Tree)를 결합하여 최종 예측을 만든다.

1. 개념

- (1) RandomForest는 다수의 결정 트리를 독립적으로 학습시켜 그 결과를 종합하여 최종 예측을 만든다.
- (2) 각 결정 트리는 데이터의 일부 샘플과 특성의 일부를 무작위로 선택하여 학습한다.

2. 구성 및 방법

- (1) **결정 트리**: 데이터의 특성을 기준으로 분할하여 학습하는 단순한 모델이다. 각 트리는 독립적으로 학습된다.
- (2) **배깅**: 여러 트리를 학습시키기 위해 데이터의 일부를 무작위로 샘플링하여 각 트리에 제공한다. 이를 통해 모델의 분산을 줄이고 과적합을 방지한다.

1. 앙상블 모델 구조

(3) 예측 종합: 각 트리의 예측 결과를 평균 내거나 다수결로 최종 예측을 결정한다. 분류 문제에서는 다수결, 회귀 문제에서는 평균을 사용한다.

3. 장점

- (1) 높은 정확도:** 여러 트리를 결합하여 예측 성능을 향상시킨다.
- (2) 과적합 방지:** 각 트리가 독립적으로 학습되므로 과적합을 방지할 수 있다.
- (3) 피쳐 중요도 평가:** 각 피쳐가 예측에 얼마나 중요한지 평가할 수 있다.
- (4) 다양한 데이터 처리:** 분류와 회귀 문제 모두에 사용할 수 있다.

1. 앙상블 모델 구조

4. 단점

- (1) 느린 예측 속도: 많은 트리를 사용하므로 예측 속도가 느릴 수 있다.
- (2) 복잡한 모델 구조: 많은 트리를 결합하므로 모델이 복잡해진다.

5. 사용 예

- (1) 금융 데이터에서 신용 위험 평가
- (2) 의료 데이터에서 질병 진단

1. 앙상블 모델 구조

부스팅(Boosting)

- 부스팅은 여러 모델을 순차적으로 학습시키며, 이전 모델이 만든 오류를 다음 모델이 수정해 나가는 방식이다.
- 각 모델은 이전 모델의 예측 오류를 기반으로 가중치를 조정하며 학습한다.
- 이를 통해 모델의 편향을 줄이고, 예측 성능을 향상시킨다.

여러 개의 약한 학습기(분류기)가 순차적으로 학습, 예측 하면서 잘못 예측한 데이터에 가중치를 부여하여 오류를 개선해 나가는 학습 방법
대표적인 부스팅 알고리즘은 XGBoost이다.

XGBoost(Kaggle 대회 상위 석권), LightGBM(속도 빠름)

(1) AdaBoost(Adaptive Boosting) : 개별 약한 학습기에 순차적으로 가중치를 부여해 결합하여 예측 , 초기 부스팅 모델

(2) GBM(Gradient Boost Machine) : 에이다 부스트와 유사하나 가중치를 경사하강법(Gradient Descent)을 사용하여 구한다. (성능은 XGBoost보다 떨어짐)

1. 앙상블 모델 구조

XGBoost 모델 소개

XGBoost는 부스팅(Boosting) 기법을 활용한 앙상블 모델로, Gradient Boosting 알고리즘을 기반으로 한다. 뛰어난 성능과 효율성을 자랑하며, 특히 대규모 데이터셋에서 효과적이다.

1. 개념

1. XGBoost는 여러 개의 약한 학습기(주로 결정 트리)를 순차적으로 학습시키며, 이전 모델의 예측 오류를 다음 모델이 수정해 나가는 방식으로 성능을 향상시키는 부스팅 알고리즘이다.
2. 최적화와 병렬 학습 기술을 통해 기존 Gradient Boosting보다 더 빠르고 효율적이다.

2. 구성 및 방법

1. **Gradient Boosting**: 기본적인 아이디어는 손실 함수를 최소화하기 위해 순차적으로 모델을 추가하여 예측 성능을 개선하는 것이다.
2. **오류 수정**: 각 단계에서 모델은 이전 모델이 만든 예측 오류를 기반으로 학습하여, 오류를 줄이도록 보완한다.
3. **정규화**: 과적합을 방지하기 위해 정규화 기법을 사용하여 모델의 복잡도를 제어한다.
4. **병렬 학습**: 트리 구조를 병렬로 구축하여 학습 속도를 크게 향상시킨다.

3. 장점

1. **높은 성능**: 복잡한 데이터셋에서도 높은 예측 성능을 발휘한다.
2. **빠른 학습 속도**: 병렬 학습과 최적화 기법을 통해 빠른 학습 속도를 제공한다.
3. **유연성**: 회귀, 분류, 랭킹 등 다양한 문제에 적용할 수 있다.
4. **과적합 방지**: 정규화 및 조기 종료 기능으로 과적합을 방지한다.

4. 단점

1. **복잡한 튜닝**: 최적의 성능을 위해 하이퍼파라미터를 세밀하게 튜닝해야 한다.
2. **메모리 사용량**: 대규모 데이터셋 처리 시 메모리 사용량이 많을 수 있다.

1. 앙상블 모델 구조

XGBoost 모델 소개

XGBoost는 부스팅(Boosting) 기법을 활용한 앙상블 모델로, Gradient Boosting 알고리즘을 기반으로 한다. 뛰어난 성능과 효율성을 자랑하며, 특히 대규모 데이터셋에서 효과적이다.

1. 개념

1. XGBoost는 여러 개의 약한 학습기(주로 결정 트리)를 순차적으로 학습시키며, 이전 모델의 예측 오류를 다음 모델이 수정해 나가는 방식으로 성능을 향상시키는 부스팅 알고리즘이다.
2. 최적화와 병렬 학습 기술을 통해 기존 Gradient Boosting보다 더 빠르고 효율적이다.

2. 구성 및 방법

1. **Gradient Boosting**: 기본적인 아이디어는 손실 함수를 최소화하기 위해 순차적으로 모델을 추가하여 예측 성능을 개선하는 것이다.
2. **오류 수정**: 각 단계에서 모델은 이전 모델이 만든 예측 오류를 기반으로 학습하여, 오류를 줄이도록 보완한다.
3. **정규화**: 과적합을 방지하기 위해 정규화 기법을 사용하여 모델의 복잡도를 제어한다.
4. **병렬 학습**: 트리 구조를 병렬로 구축하여 학습 속도를 크게 향상시킨다.

3. 장점

1. **높은 성능**: 복잡한 데이터셋에서도 높은 예측 성능을 발휘한다.
2. **빠른 학습 속도**: 병렬 학습과 최적화 기법을 통해 빠른 학습 속도를 제공한다.
3. **유연성**: 회귀, 분류, 랭킹 등 다양한 문제에 적용할 수 있다.
4. **과적합 방지**: 정규화 및 조기 종료 기능으로 과적합을 방지한다.

4. 단점

1. **복잡한 튜닝**: 최적의 성능을 위해 하이퍼파라미터를 세밀하게 튜닝해야 한다.
2. **메모리 사용량**: 대규모 데이터셋 처리 시 메모리 사용량이 많을 수 있다.

1. 앙상블 모델 구조

XGBoost 모델 소개

XGBoost는 부스팅(Boosting) 기법을 활용한 앙상블 모델로, Gradient Boosting 알고리즘을 기반으로 한다. 뛰어난 성능과 효율성을 자랑하며, 특히 대규모 데이터셋에서 효과적이다.

1. 개념

1. XGBoost는 여러 개의 약한 학습기(주로 결정 트리)를 순차적으로 학습시키며, 이전 모델의 예측 오류를 다음 모델이 수정해 나가는 방식으로 성능을 향상시키는 부스팅 알고리즘이다.
2. 최적화와 병렬 학습 기술을 통해 기존 Gradient Boosting보다 더 빠르고 효율적이다.

2. 구성 및 방법

1. **Gradient Boosting**: 기본적인 아이디어는 손실 함수를 최소화하기 위해 순차적으로 모델을 추가하여 예측 성능을 개선하는 것이다.
2. **오류 수정**: 각 단계에서 모델은 이전 모델이 만든 예측 오류를 기반으로 학습하여, 오류를 줄이도록 보완한다.
3. **정규화**: 과적합을 방지하기 위해 정규화 기법을 사용하여 모델의 복잡도를 제어한다.
4. **병렬 학습**: 트리 구조를 병렬로 구축하여 학습 속도를 크게 향상시킨다.

3. 장점

1. **높은 성능**: 복잡한 데이터셋에서도 높은 예측 성능을 발휘한다.
2. **빠른 학습 속도**: 병렬 학습과 최적화 기법을 통해 빠른 학습 속도를 제공한다.
3. **유연성**: 회귀, 분류, 랭킹 등 다양한 문제에 적용할 수 있다.
4. **과적합 방지**: 정규화 및 조기 종료 기능으로 과적합을 방지한다.

4. 단점

1. **복잡한 튜닝**: 최적의 성능을 위해 하이퍼파라미터를 세밀하게 튜닝해야 한다.
2. **메모리 사용량**: 대규모 데이터셋 처리 시 메모리 사용량이 많을 수 있다.

2. 앙상블 모델 구현 실습

2. 앙상블 모델 구현 실습

Mushroom.csv 데이터 파일 엑셀로 확인



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	p	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w
2	e	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w
3	e	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w
4	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w
5	e	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w
6	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w
7	e	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w
8	e	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w
9	p	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w
10	e	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w
11	e	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w
12	e	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w
13	e	b	s	y	t	a	f	c	b	w	e	c	s	s	w	w
14	p	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w
15	e	x	f	n	f	n	f	w	b	n	t	e	s	f	w	w
16	e	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w
17	e	f	f	w	f	n	f	w	b	k	t	e	s	s	w	w

2. 앙상블 모델 구현 실습

mushroom.csv
스프레드시트



데이터 분석 결과를 요약하면 다음과 같습니다:

1. 데이터 개요

- 총 행(Row) 수: 8123
- 총 열(Column) 수: 23
- 모든 열이 결측값 없이 존재함

2. 각 열(Column)의 데이터 타입

- 모든 열은 문자열 데이터(``object``) 타입으로 구성됨

3. 기본 통계

- 각 열의 고유값(unique) 개수, 최빈값(top), 최빈값 빈도수(freq)를 제공함

4. 특정 열의 주요 특징

- ``p``: 두 개의 고유값이 있으며, 최빈값은 'e', 빈도수는 4208
- ``x``: 여섯 개의 고유값이 있으며, 최빈값은 'x', 빈도수는 3655
- ``s``: 네 개의 고유값이 있으며, 최빈값은 'y', 빈도수는 3244
- ``n``: 열 개의 고유값이 있으며, 최빈값은 'n', 빈도수는 2283
- ``t``: 두 개의 고유값이 있으며, 최빈값은 'f', 빈도수는 4748

2. 앙상블 모델 구현 실습

Kaggle에서 검색해서 mushroom csv 파일의 각 컬럼을 설명해줘.



사이트 6개 검색함

mushroom 데이터셋의 각 컬럼에 대한 설명은 다음과 같습니다:

1. **class:** 버섯의 독성 여부를 나타냅니다. 두 가지 값이 있습니다.
 - e: 식용
 - p: 독성
2. **cap-shape:** 버섯 갓의 형태입니다.
 - bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
3. **cap-surface:** 버섯 갓의 표면입니다.
 - fibrous=f, grooves=g, scaly=y, smooth=s
4. **cap-color:** 버섯 갓의 색상입니다.
 - brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
5. **bruises:** 버섯에 멍이 있는지 여부입니다.
 - bruises=t, no=f

2. 앙상블 모델 구현 실습

6. **odor**: 버섯의 냄새입니다.

- almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s

7. **gill-attachment**: 버섯 주름의 부착 방식입니다.

- attached=a, descending=d, free=f, notched=n

8. **gill-spacing**: 버섯 주름의 간격입니다.

- close=c, crowded=w, distant=d

9. **gill-size**: 버섯 주름의 크기입니다.

- broad=b, narrow=n

10. **gill-color**: 버섯 주름의 색상입니다.

- black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y

11. **stalk-shape**: 버섯 자루의 형태입니다.

- enlarging=e, tapering=t

12. **stalk-root**: 버섯 자루의 뿌리 형태입니다.

- bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?

2. 앙상블 모델 구현 실습

13. **stalk-surface-above-ring**: 버섯 자루의 위쪽 표면입니다.

- fibrous=f, scaly=y, silky=k, smooth=s

14. **stalk-surface-below-ring**: 버섯 자루의 아래쪽 표면입니다.

- fibrous=f, scaly=y, silky=k, smooth=s

15. **stalk-color-above-ring**: 버섯 자루의 위쪽 색상입니다.

- brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

16. **stalk-color-below-ring**: 버섯 자루의 아래쪽 색상입니다.

- brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

17. **veil-type**: 버섯 덮개 유형입니다.

- partial=p, universal=u

18. **veil-color**: 버섯 덮개의 색상입니다.

- brown=n, orange=o, white=w, yellow=y

2. 앙상블 모델 구현 실습

19. **ring-number**: 버섯 자루에 있는 고리의 수입니다.

- none=n, one=o, two=t

20. **ring-type**: 버섯 자루 고리의 유형입니다.

- cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z

21. **spore-print-color**: 포자 인쇄 색상입니다.

- black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y

22. **population**: 버섯 개체군의 크기입니다.

- abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y

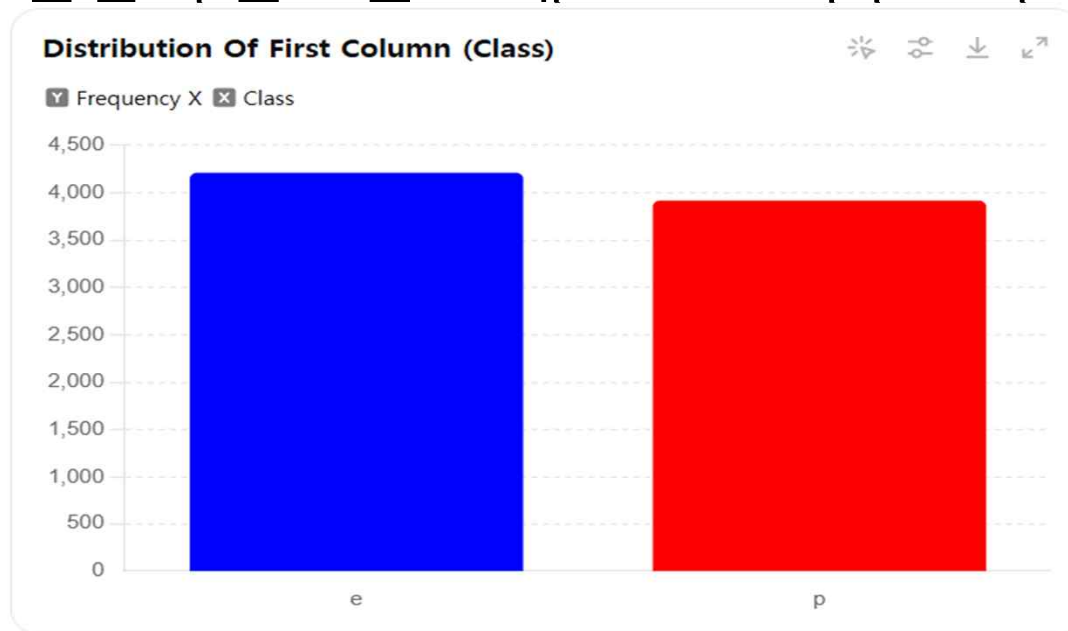
23. **habitat**: 버섯의 서식지입니다.

- grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

이 정보는 주로 Kaggle과 UCI Machine Learning Repository에서 제공되었습니다 ([Kaggle](#)) ([Kaggle](#)) ([UCI Machine Learning Repository](#)) ([DataScience Plus](#)).

2. 앙상블 모델 구현 실습

첫 번째 컬럼의 분포를 그래프로 그려주고 수치로도 알려줘.



첫 번째 컬럼의 분포는 다음과 같습니다:

- 식용(e): 4208개
- 독성(p): 3915개

위의 그래프는 이 분포를 시각적으로 나타낸 것입니다. 식용 버섯이 약간 더 많음을 알 수 있습니다.

[>-]



2. 앙상블 모델 구현 실습(RandomForest)

RandomForest를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.



랜덤 포레스트를 사용한 이진 분류 모델의 성능 평가지표는 다음과 같습니다:

- 정확도(Accuracy): 1.0
- 정밀도(Precision), 재현율(Recall), F1-점수(F1-score):
 - 클래스 'e' (식용 버섯):
 - 정밀도: 1.00
 - 재현율: 1.00
 - F1-점수: 1.00
 - 지원(Support): 853
 - 클래스 'p' (독성 버섯):
 - 정밀도: 1.00
 - 재현율: 1.00
 - F1-점수: 1.00
 - 지원(Support): 772
- 전체:
 - 정확도: 1.00
 - 매크로 평균(Macro avg) 정밀도: 1.00, 재현율: 1.00, F1-점수: 1.00
 - 가중 평균(Weighted avg) 정밀도: 1.00, 재현율: 1.00, F1-점수: 1.00

모델이 테스트 데이터에서 모든 샘플을 정확하게 분류한 것으로 나타났습니다. 이는 모델이 매우 높은 성능을 보였음을 의미합니다. [↗]



2. 앙상블 모델 구현 실습(XGBoost)

XGBoost를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

XGBoost를 사용한 이진 분류 모델의 성능 평가지표는 다음과 같습니다:

- **정확도(Accuracy):** 1.0
- **정밀도(Precision), 재현율(Recall), F1-점수(F1-score):**
 - 클래스 '0' (식용 버섯):
 - 정밀도: 1.00
 - 재현율: 1.00
 - F1-점수: 1.00
 - 지원(Support): 853
 - 클래스 '1' (독성 버섯):
 - 정밀도: 1.00
 - 재현율: 1.00
 - F1-점수: 1.00
 - 지원(Support): 772
- **전체:**
 - 정확도: 1.00
 - 매크로 평균(Macro avg) 정밀도: 1.00, 재현율: 1.00, F1-점수: 1.00
 - 가중 평균(Weighted avg) 정밀도: 1.00, 재현율: 1.00, F1-점수: 1.00

이 모델도 테스트 데이터에서 모든 샘플을 정확하게 분류한 것으로 나타났습니다. 이는 모델이 매우 높은 성능을 보였음을 의미합니다. [↗]

2. 앙상블 모델 구현 실습(모델 성능비교)



Breast Cancer.csv

스프레드시트


📎 결정 트리를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

결정 트리를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

이진 분류 모델에 대한 결정 트리의 평가지표는 다음과 같습니다:

- 정확도 (Accuracy): 94.74%
- 정밀도 (Precision): 95.77%
- 재현율 (Recall): 95.77%
- F1 점수 (F1 Score): 95.77%
- 혼동 행렬 (Confusion Matrix):
 - 참 음성 (TN): 40
 - 거짓 양성 (FP): 3
 - 거짓 음성 (FN): 3
 - 참 양성 (TP): 68

2. 앙상블 모델 구현 실습(모델 성능비교)

 Breast Cancer.csv
스프레드시트

 RandomForest를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

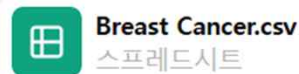
RandomForest를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.



RandomForest 이진 분류 모델의 평가지표는 다음과 같습니다:

- **정확도 (Accuracy):** 0.9649
- **정밀도 (Precision):** 0.9589
- **재현율 (Recall):** 0.9859
- **F1 점수 (F1 Score):** 0.9722
- **ROC AUC:** 0.9953
- **혼동 행렬 (Confusion Matrix):**
 - 참 부정 (True Negatives): 40
 - 거짓 긍정 (False Positives): 3
 - 거짓 부정 (False Negatives): 1
 - 참 긍정 (True Positives): 70

2. 앙상블 모델 구현 실습(모델 성능비교)



📎 XGBoost를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

XGBoost를 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.



다음은 XGBoost 이진 분류 모델에 대한 평가 지표입니다:

- **정확도 (Accuracy):** 95.61%
- **정밀도 (Precision):** 95.83%
- **재현율 (Recall):** 97.18%
- **F1 점수 (F1 Score):** 96.50%
- **ROC AUC:** 99.31%

이 지표들은 모델이 목표 변수를 예측하는 데 매우 우수한 성능을 보이고 있음을 나타냅니다. 추가적인 분석이나 수정이 필요하면 말씀해 주세요!



2. 앙상블 모델 구현 실습(모델 성능비교)


Windows의 cmd를 실행하여 명령창에서 LightGBM을 설치한다
“pip3 instal lightgbm”


```
명령 프롬프트
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\storm>pip3 install lightgbm
Defaulting to user installation because normal site-packages is not writeable
Collecting lightgbm
  Downloading lightgbm-4.4.0-py3-none-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.17.0 in c:\users\storm\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from lightgbm) (1.26.4)
Requirement already satisfied: scipy in c:\users\storm\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from lightgbm) (1.13.1)
Downloading lightgbm-4.4.0-py3-none-win_amd64.whl (1.4 MB)
  1.4/1.4 MB 8.3 MB/s eta 0:00:00
Installing collected packages: lightgbm
Successfully installed lightgbm-4.4.0

C:\Users\storm>python3
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import lightgbm
```

2. 앙상블 모델 구현 실습(모델 성능비교)

 Breast Cancer.csv
스프레드시트

 LightGBM을 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

LightGBM을 사용한 이진분류 모델 만들어주고 평가지표를 알려줘.

실행 오류!!

: scikit-learn과 LightGBM 버전이 매치가 되지 않음..



현재 외부 패키지를 설치할 수 없으므로, 로컬 환경에서 LightGBM 패키지를 설치하고 사용해야 합니다. 아래는 이진분류 모델을 만들고 평가하는 전체 코드입니다. 이 코드를 로컬 환경에서 실행해 주세요.

2. 앙상블 모델 구현 실습(모델 성능비교)

결론

BreastCancer 데이터 셋에 대한 모델의 정확도는

결정 트리 < XGBoost < RandomForest

의 순서로 성능이 높다

감사합니다