bagging, boosting

부산대학교 정보컴퓨터공학부

15 학번 이연걸

1. 개요

Decision Tree 를 이용하여 와인 데이터를 분류하는 과제를 진행한다. 앙상블을 적용하지 않은 모델, bagging 을 적용한 모델, boosting 을 적용한 모델 등으로 나누어 진행한 뒤, 도출된 결과와 특징을 비교한다.

2. 구현 과정

앙상블을 적용하지 않은 모델, bagging 을 적용한 모델, boosting 을 적용한 모델 순으로 실험을 진행한다.

2.1 앙상블을 적용하지 않은 모델

max_depth=1 으로 적용하고 진행한다. 희석 와인의 OD280/OD315 비율이 2.19 보다 작거나 같은가와 큰가가 기준이 된다.

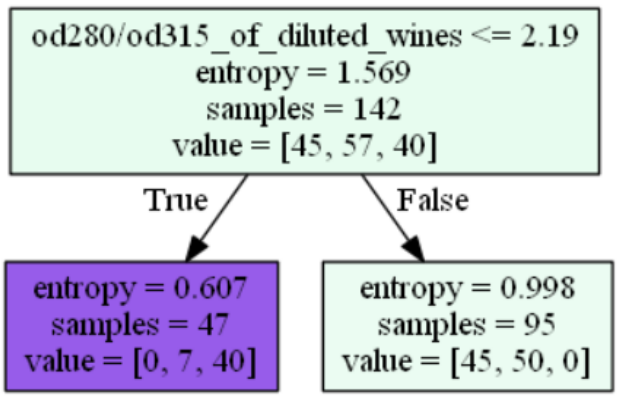

그림 1 – 앙상블을 적용하지 않은 모델

표 1 – 앙상블을 적용하지 않은 모델

	precision	recall	fl-score	support
0.0	0.00	0.00	0.00	14
1.0	0.43	0.86	0.57	14
2.0	0.75	0.75	0.75	8
accuracy	N/A	N/A	0.50	36
macro avg	0.39	0.54	0.44	36
weighted avg	0.33	0.50	0.39	36

test

2.2 bagging 을 적용한 모델

bagging 이라는 이름은 Bootstrap AGGregation 에서 나온 것이다. 먼저 데이터에서 일부를 샘플링 하는 부트스트랩(bootstrap)을 실시한 뒤, 이렇게 만든 각각의 부트스트랩 샘플에 모형을 각각 학습시킨다. 그러면 하나의 데이터셋에서 여러 개의 샘플을 만들어 모형을 여러 번 학습시킬 수 있다. 이들의 예측을 합쳐 최종 예측을 한다.

bagging 은 Decision Tree 와 같이 불안정한 모형에 좋으며, 대표적으로 랜덤 포레스트가 있다. 이는 의사결정나무에 배깅을 적용한 모형으로 각각의 나무는 표준적인 배깅과 달리 무작위로 선택된 특성을 이용한다. 이를 특성 배깅(feature bagging)이라고 한다. 부트스트랩을 해서 샘플마다 차이가 있어도 예측력이 높은 특성이 있으면 모든 나무들이 같은 특성을 사용하기 때문에 앙상블을 하는 의의가 없다. 그래서 나무에서 사용할 특성을 무작위로 선택해서 다양한 나무를 만드는 것이다.

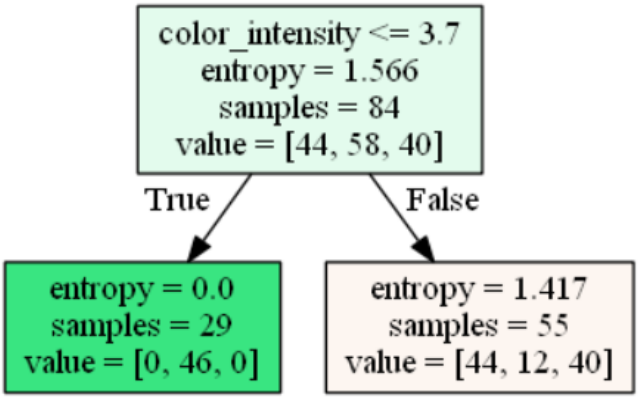

그림 2 – bagging 을 적용한 모델 (random forest)

표 2 – bagging 을 적용한 모델 (random forest)

	precision	recall	fl-score	support
0.0	0.88	1.00	0.93	14
1.0	0.72	1.00	0.82	14
2.0	0.00	0.00	0.00	8
accuracy	N/A	N/A	0.78	36
macro avg	0.53	0.67	0.59	36
weighted avg	0.61	0.78	0.68	36

2.3 boosting 을 적용한 모델

boosting 은 머신러닝 앙상블 기법 중 하나로 sequential 한 weak learner 들을 여러 개 결합하여 예측 혹은 분류 성능을 높이는 알고리즘이다. 결구 overfitting 을 막기 위해 약한 모델을 여러 개 결합시켜 그 결과를 종합하는 것이 기본적인 아이디어이며, 부스팅은 여기에 seauential 이 추가된다.

즉 연속적인 weak learner, 바로 직전 weak learner 의 error 를 반영한 현재 weak learner 를 잡겠다는 아이디어이다. 부스팅 계열 모델은 AdaBoost, GBM, XGBoost, LightGBM, CatBoost 등이 있으며 최초로 개발된 AdaBoost 와 더 발전된 GBM 을 사용해본다.

2.3.1 AdaBoost 를 적용한 모델

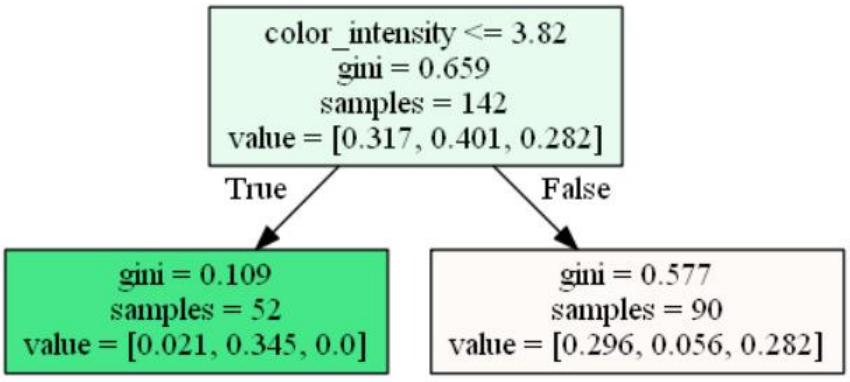

그림 3 – AdaBoost 를 적용한 모델

표 3 - AdaBoost 를 적용한 모델

	precision	recall	fl-score	support
0.0	0.54	0.93	0.68	14
1.0	0.92	0.79	0.85	14
2.0	0.00	0.00	0.00	8
accuracy	N/A	N/A	0.67	36
macro avg	0.49	0.57	0.51	36
weighted avg	0.57	0.67	0.60	36

2.3.2 GBM 을 적용한 모델

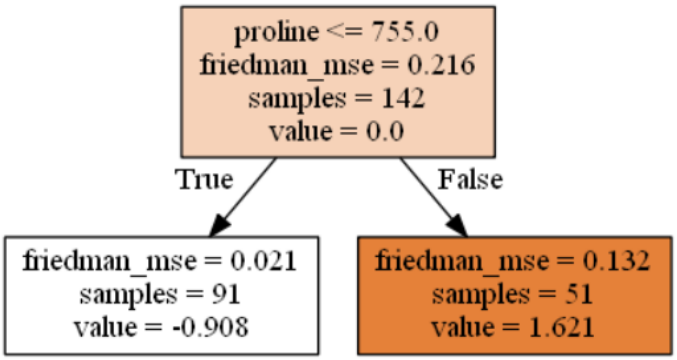

그림 4 – GBM 을 적용한 모델

표 4 – GBM 을 적용한 모델

	precision	recall	fl-score	support
0.0	0.87	0.93	0.90	14
1.0	0.67	1.00	0.80	14
2.0	0.00	0.00	0.00	8
accuracy	N/A	N/A	0.75	36
macro avg	0.51	0.64	0.57	36
weighted avg	0.60	0.75	0.66	36

3. 총평

앙상블을 적용한 경우 대체적으로 앙상블을 적용하지 않은 경우보다 정확도 등의 수치가 더 높게 나왔다. 배깅의 경우 몇번 돌려봤을 때 정확도 수치가 계속 차이를 보였다. 이를 통해 상대적으로 부스팅에 비해 error 가 많음을 알 수 있었다. 하지만, 부스팅의 경우 상대적으로 성능은 좋지만 이 때문에 오버 피팅이 될 가능성이 높아진다.

따라서 Decision Tree 를 분석한 뒤, 성능이 낮다면 부스팅을 오버 피팅이 문제라면 배깅을 적용하는 식으로 진행해야 할 것이다.