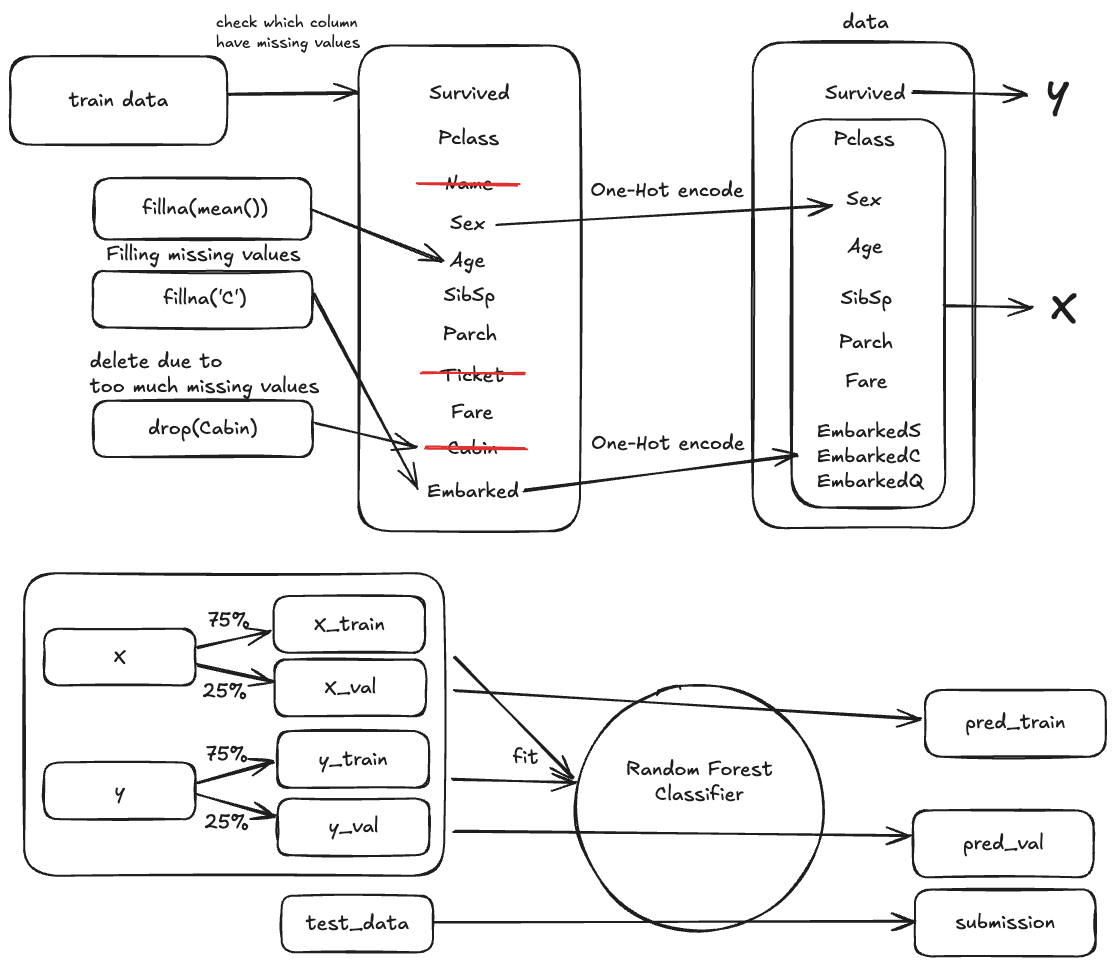
2주차 결과 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20221559 이름: 박준우

1. 

1) train data를 가져와서, column별 비어있는 값의 백분율을 출력해 알아봤다.

2) 연령, 승선위치, 탑승 선실 정보의 결측값을 확인했고, 연령은 평균값으로, 승선위치는 가장 많은 값으로 채우고, 탑승 선실 정보는 결측값이 너무 많아 삭제했다.

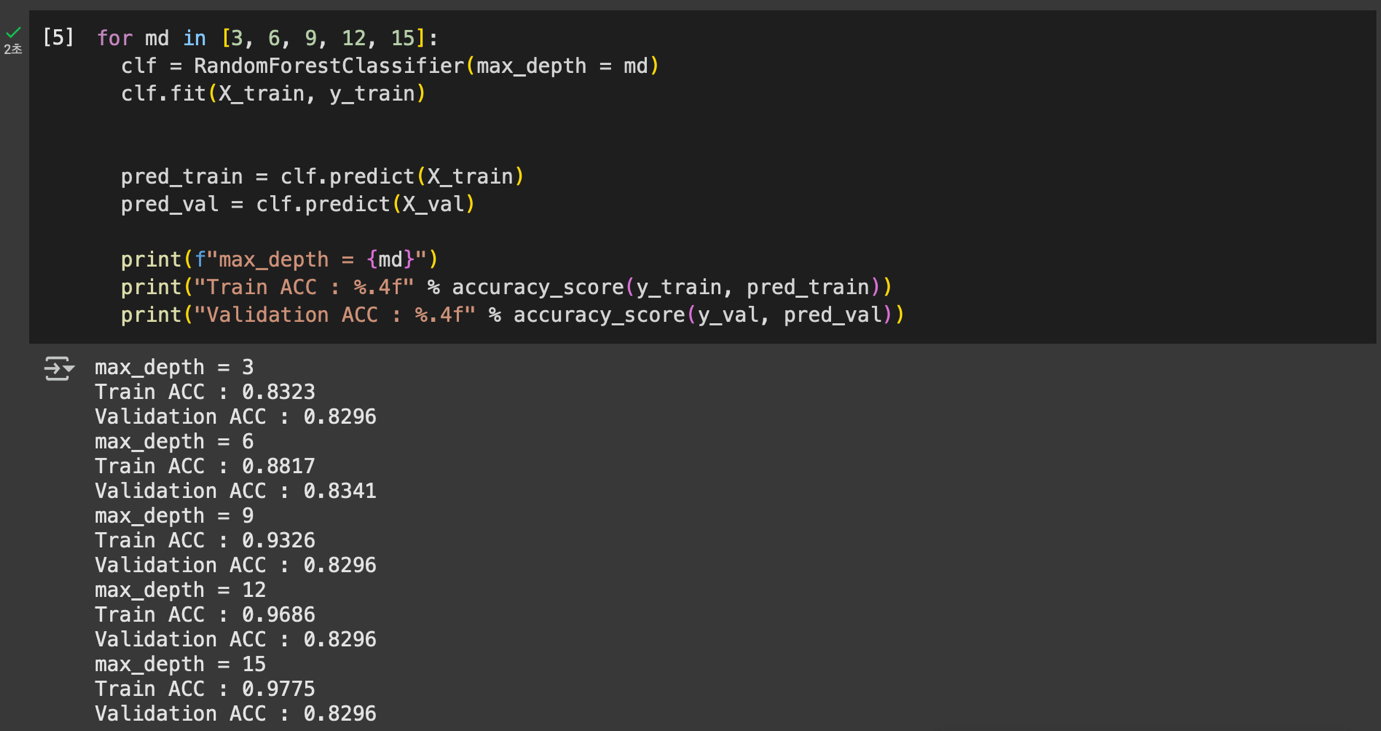
3) 이름과 티켓번호는 쓸모가 없다 판단하여 삭제했다.

4) 성별 정보, 승선위치가 수치정보가 아니어서, one-hot encode 했다.

5) train data를 학습용, 확인용으로 나누어서 학습시키고, 평가한다.

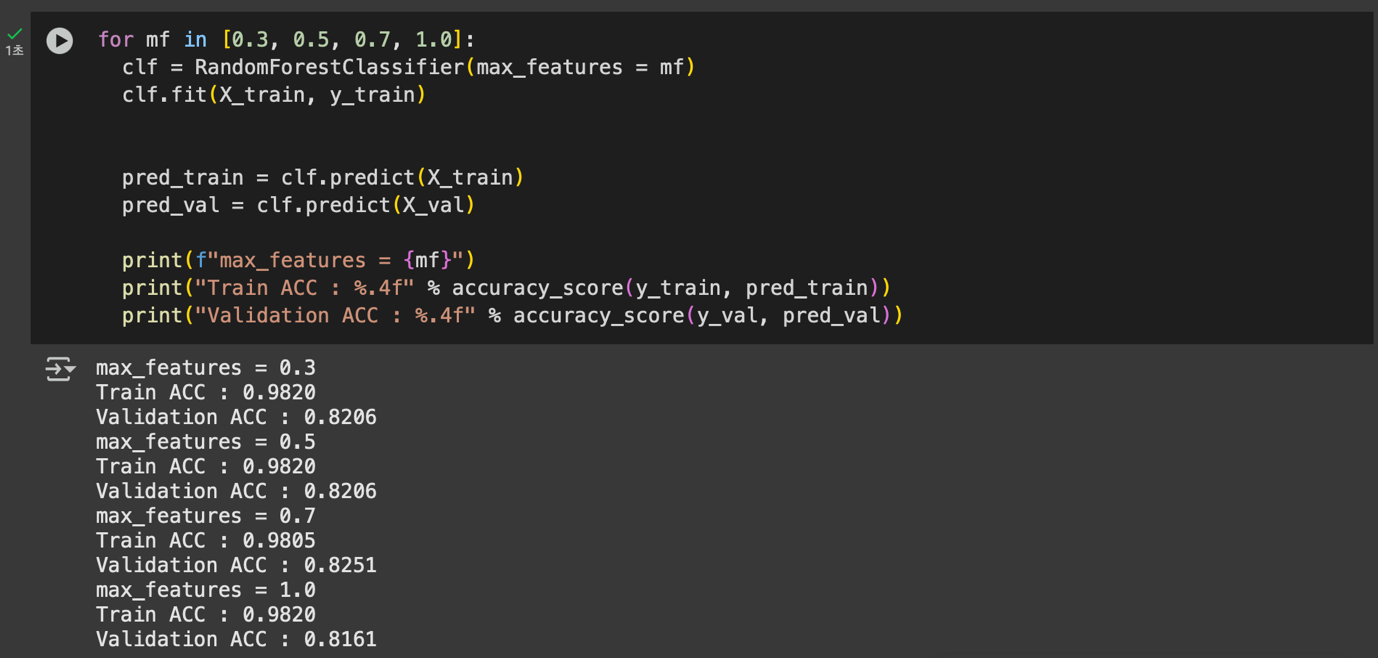
6) 테스트 데이터에 학습용 데이터에 했던 처리를 동일하게 하여 결과를 만들어 제출한다.

2. max\_depth = [3, 6, 9, 12, 15], max\_features = [0.3, 0.5 0.7, 1.0], n\_estimators = [50, 75, 100, 250, 1000]으로, 나머지 하이퍼파라미터는 기본으로 진행하겠다.



Max\_depth를 바꿔가면서 진행했다.

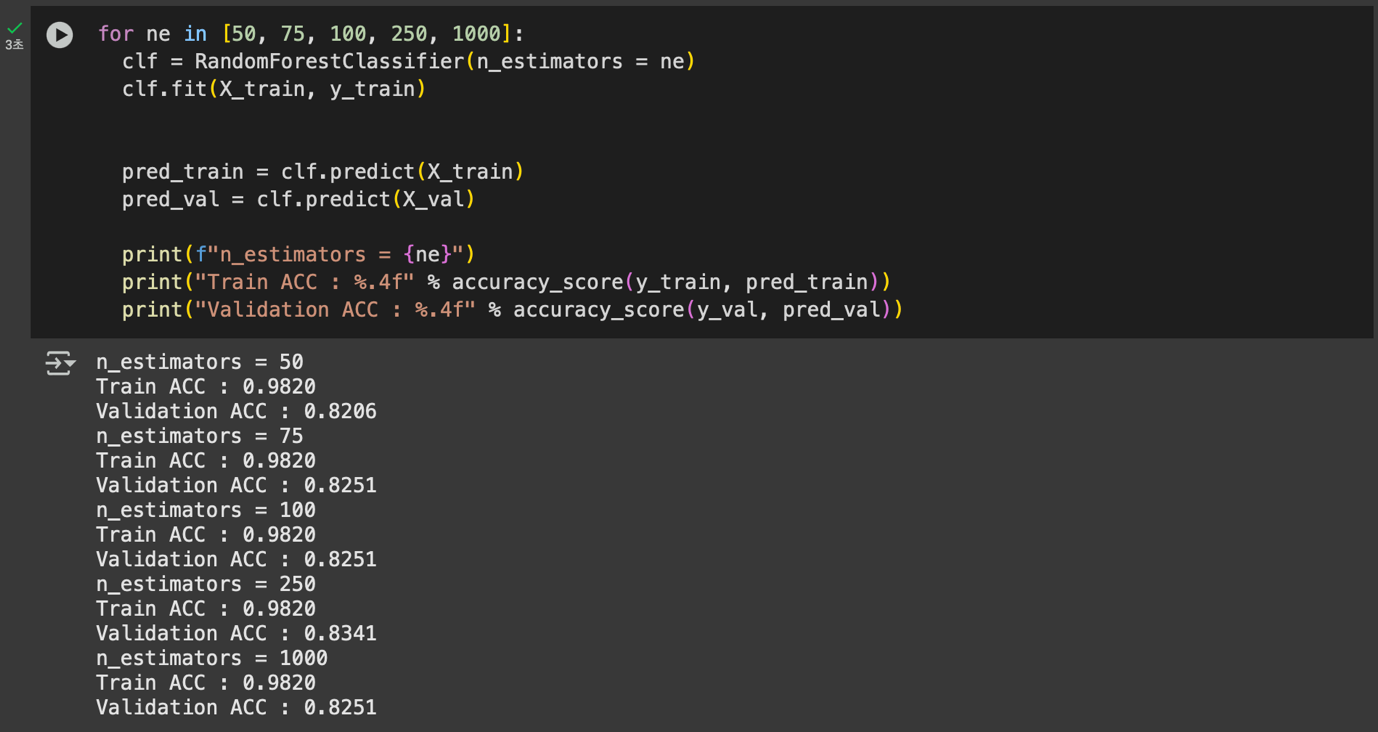
깊이가 깊어질수록 학습 데이터에는 잘 최적화 되었지만, 오히려 검증 데이터는 6에서보다 잘 맞추지 못하는 모습이다. 과하게 최적화되지 않기 위해서, 적당히 제한하면 도움이 되는 것으로 보인다.



Max\_features를 조정해가며 진행했다.

Random Forest는 전체 feature 중 일부씩만 뽑아 트리를 만드는데, 1.0이라 함은 일부를 뽑는 과정이 없이 전체를 쓰는 것이라, 검증 데이터의 정확도가 떨어지는 모습을 보인다.

해당 데이터는 0.7에서 가장 검증 데이터 정확도가 높았다.



N\_estimators(만들 의사 결정 트리 개수) 를 조정해가며 만들었다.

학습 데이터의 결과는 이 수를 조정해도 맞춘 결과의 개수가 같다. 다만, 250개에서 검증 데이터에서의 정답률이 소폭 올랐다.

3. Random Forest는 의사 결정 트리를 기반으로 만들어지는 모델이다. 이는 주어진 데이터들을 설명할 수 있는 의사 결정 트리를 만들어서, 새로운 데이터를 해당 트리에 넣어서 결과를 예측하기 때문에, 학습 데이터에서의 정확도가 제출시의 정확도보다 어떻게 보면 높을 수 밖에 없는 구조인 것 같다.

따라서 트리를 촘촘하게 만들어서 정확도를 올릴지, 혹은 이것이 자료에 과적합을 유도한다고 생각해 오히려 트리를 작게 만들어 전체의 경향을 반영하도록 할지 사이의 적당한 균형을 찾아 테스트 정확도를 올리는 것이 모델의 최적화 목표가 된다.