2주차 과제

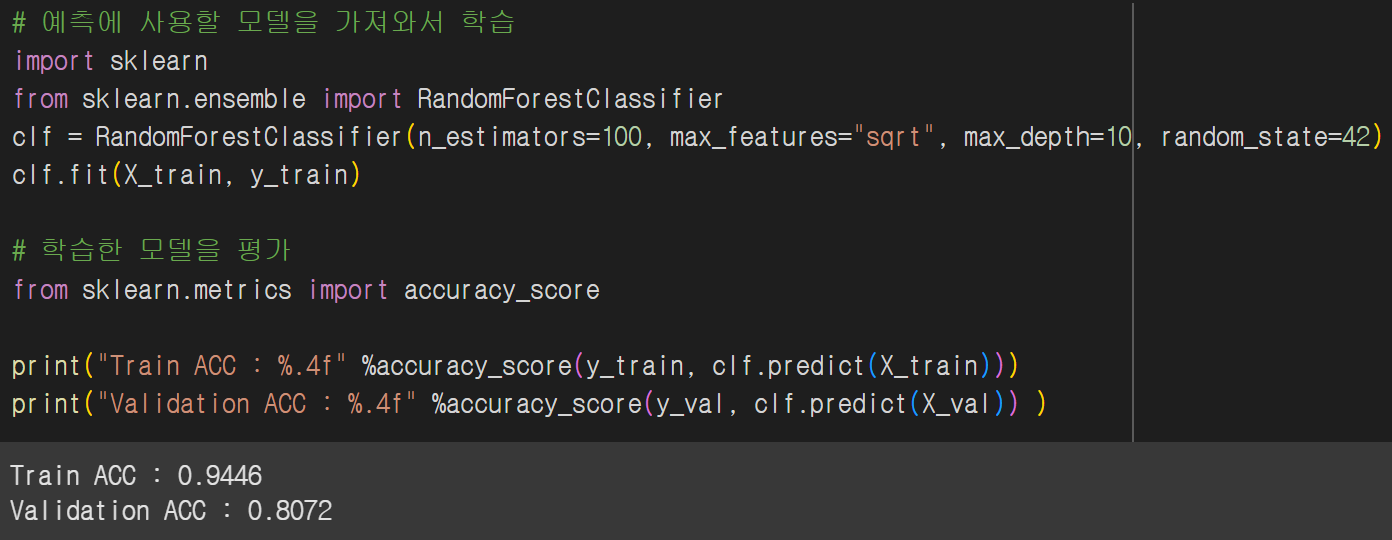
전공: 컴퓨터공학과 학년: 2학년 학번: 20221519 이름: 구현모

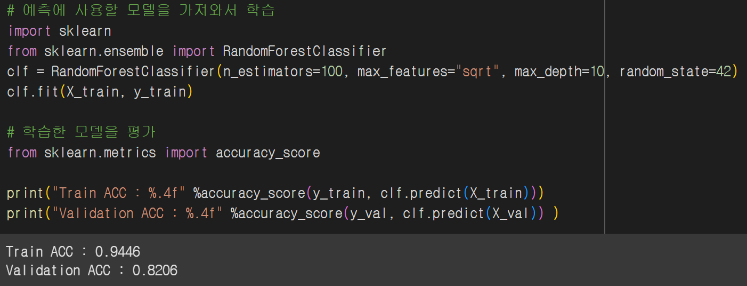
1. Q4에서 One-Hot encoding 대신에, Ordinal Encoding을 수행하고 나서 나머지는 동일한 조건으로 학습을 했을 때 Train, Validation Accuracy를 계산해보세요.

텍스트, 스크린샷, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진

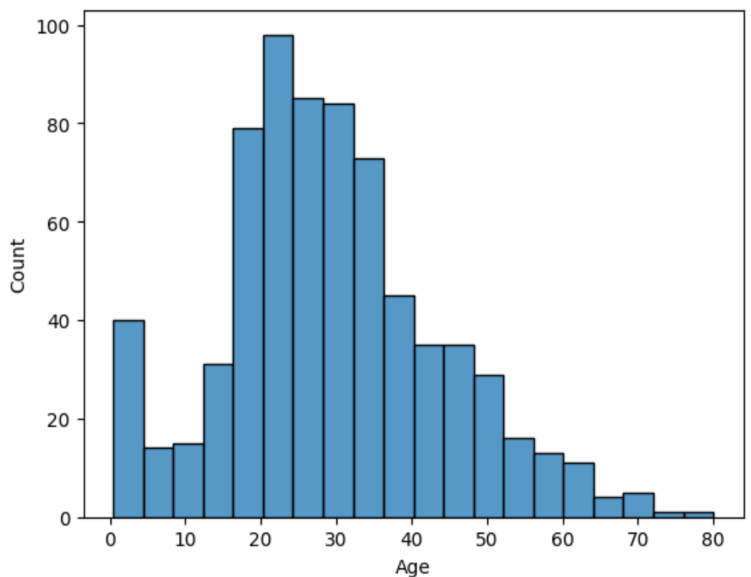
AI가 생성한 콘텐츠는 부정확할 수 있습니다. 다음은 Ordinal Encoding으로 범주형 데이터인 Sex, Embarked column을 수치형 데이터로 변환한 코드이다.

Ordinal Encoding이 되어서 다음과 같이 데이터가 Sex의 경우는 male, female에서 1.0, 0.0으로, Embarked의 경우는 S, Q, C에서 2.0, 1.0, 0.0으로 변환되었다.

위에 있는 사진은 One-Hot encoding을 수행한 데이터를 기반으로 학습한 모델의 Train, Validation Accuracy 값이고 밑에 있는 사진은, Ordinal Encoding을 진행한 데이터를 바탕으로 학습한 모델의 Train, Validation Accuracy 값이다. 두 모델은 인코딩 방식을 제외하면 똑 같은 조건에서 실행되었다.

One-Hot encoding의 경우에는 Train Accuracy가 0.9446, Validation Accuracy가 0.8072가 나왔으며 Ordinal Encoding의 경우는 Train Accuracy가 0.9446, Validation Accuracy가 0.8206이 나왔다. 두 경우 모두 Train Accuracy은 0.9446으로 똑같이 나왔지만 Validation Accuracy의 경우 0.8072에서 0.8206로 Ordinal Encoding의 경우에 성능이 증가했음을 확인할 수 있다.

2. Age에 대한 정보를 그냥 중앙값으로 채웠는데, 생각해보니 Age값 자체보다 나이대별로 구분한 정보가 더 도움이 많이 될 것 같다는 판단을 했습니다. 결측치를 채우지 않은 상태에서 나이에 대한 histogram을 그린 뒤에 적절한 구간을 정해서 4개의 서로 다른 나이대로 데이터를 변환하세요. 이 때 구간을 설정하는 것은 histogram을 보고 분석가의 의견에 따라 결정하세요. 그 이유도 함께 서술하세요.

 다음은 결측치를 채우지 않은 Age값에 대한 히스토그램이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.구간은 0~12세, 13~35세, 36~55세, 56세 이상으로 설정하겠다.

0~13세를 첫번째 구간으로 설정한 이유는 당시의 영국 초등교육법상으로 특정 조건을 만족하지 못한 13세 미만의 어린이를 고용하는 것이 불법이었기 때문에 0~12세의 경우 경제권이 없는 어린 아이라고 볼 수 있다. 이러한 어린아이의 경우 보호자의 존재나 보호받는 존재이기 때문이다. 또한 히스토그램 상에서 약 12세를 기준으로 감소하다가 증가로 바뀌기 때문에 첫번째 구역으로 설정하였다.

두번째 구간으로 13~35세를 설정한 이유는 활발히 경제활동을 하면서 활동력이 강한 청년들의 나이대이기 때문이다. 또한 히스토그램을 보면 13~35세의 젊은 성인들이 많은 편임을 확인할 수 있기 때문에 두번째 구간으로 설정하였다.

36~55세는 히스토그램 상에서 점차 인원이 줄어들고 있다. 또한 어느 정도의 경제적 여유가 생기는 나이이고 보호자의 역할을 주로 수행할 만한 나이대이기 때문에 세번째 구간으로 나누었다.

마지막으로 56세 이상으로 나눈 이유는 히스토그램 상으로 가장 인원들이 적고 그 이전의 구간과 비교했을 때 크게 변화했기 때문이다. 또한 노년의 나이이므로 활동성이나 사회적인 배려의 정도가 다른 구간과 차이가 날 것으로 예상되기 때문에 네번째 구역으로 나누었다.

3.  RandomForestClassifier를 GridSearchCV 함수를 사용해서 hyper-parameter tuning을 수행하세요. hyper-parameter tuning을 수행할 파라미터는 max\_depth, max\_feature, n\_estimators를 포함하여 5가지 이상 정해보세요. 어떤 값을 사용할지는 제한이 없습니다. 어떤 조합에서 성능이 가장 잘 나왔는지 정리해보세요.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. GridSearchCV 함수를 이용하기 위하여 다음과 같이 설정해주었다.

이 경우에 max\_depth는 [5, 10, 15], max\_features는 ['sqrt', 'log2', 0.5], n\_estimators는 [50, 100, 300, 500], min\_samples\_split는 [2, 5, 10], min\_samples\_leaf: [1, 2, 4]로 설정하여 GridSearchCV 함수를 이용하여 교차 검증하였다.

그 후 GridSearchCV를 이용하여 각 조합을 평가하였다.

max\_depth은 의사결정 트리의 최대 깊이를 결정하는 파라미터이다.

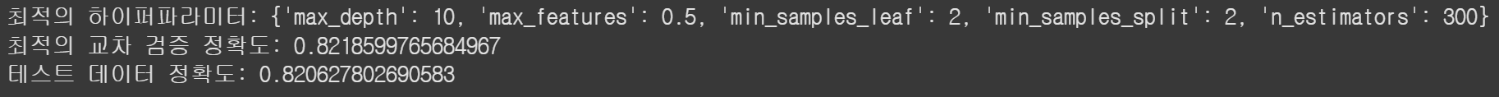
max\_features는 각 노드에서 분할을 고려할 때 사용할 최대 특성 수를 지정한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.n\_estimators는 랜덤 포레스트를 구성하는 의사결정 트리의 개수를 지정한다.

min\_samples\_split는 노드를 분할하기 위해 필요한 최소 샘플 수를 지정한다.

min\_samples\_leaf는 리프 노드에 있어야 하는 최소 샘플 수를 지정한다.

다음의 코드를 통하여 밑의 사진과 같은 결과를 얻을 수 있었다.

GridSearchCV에 따르면 최적의 하이퍼 파라미터는 max\_depth가 10, max\_features가 0.5, min\_samples\_leaf가 2, min\_samples\_split가 2, n\_estimators가 300일 때이다.

 밑의 사진은 각 파라미터들을 직접 넣어서 훈련시킨 모델을 가지고 평가를 진행한 코드와 그 결과이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

실습에서 max\_depth를 10으로 지정하고 다른 파라미터들을 기본값으로 놔준 것에 비하여 Train ACC는 조금 떨어졌지만 Validation ACC의 경우 0.8072에서 0.8206으로 올랐다는 것을 확인할 수 있다. 텍스트, 폰트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

하지만 max\_depth가 10, max\_features가 sqrt, min\_samples\_leaf가 2, min\_samples\_split가 2, n\_estimators가 300일 때 Validation ACCdml 값은 모델의 성능이 더욱 잘 나왔다.

이러한 이유는 GridSearchCV 특성 때문일 수 있다. GridSearchCV는 교차검증을 한다. cv파라미터 만큼의 교차 검증을 진행한다. 이 과정에서 데이터의 일부만 학습에 사용됨으로 전체 데이터로 학습하는 밑의 코드보다 덜 적합되었을 수 있다. 또한 교차 검증을 통하여 일반화하였고 밑의 코드의 경우 그렇지 않기 때문에 우연하게 데이터에 잘 맞는 파라미터일 수 있다. 이 경우에는 새로운 데이터에 대하여 일반화된 성능을 보장하지 못한다.

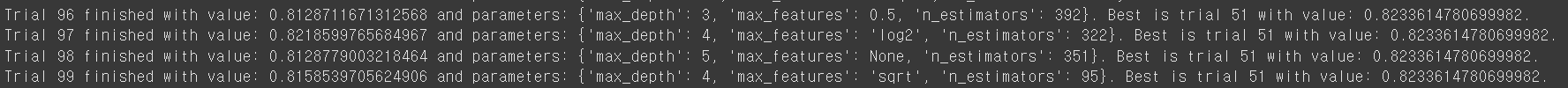
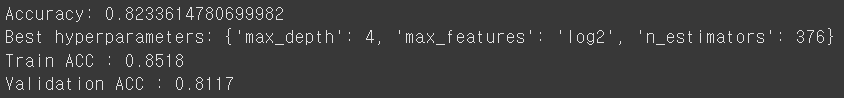
실제로 GridSearchCV로 찾은 파라미터를 이용하는 것이 Train ACC와 Validation ACC의 차이가 거의 없게 나타났다. 일반화가 잘 이루어 졌음을 알 수 있다.

더 많은 양의 데이터를 사용하여 학습을 한다면 GridSearchCV로 찾은 파라미터의 성능이 더 좋아질 가능성이 높다.

4. 실제 프로젝트를 수행할 때는 hyper-parameter tuning을 편리하게 사용할 수 있는 오픈소스 라이브러리를 사용합니다. **optuna**는 가장 많이 사용되는 Auto Hyper-parameter tuning framework입니다. 아래 웹페이지에서 Code Example을 참고하여 RandomForestClassifier의 max\_depth, max\_features, n\_estimators를 자동으로 hyper-parameter tuning하는 코드를 작성하고 그 결과를 정리해보세요.

텍스트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.다음과 같이 코드를 구현하였다.



다음과 같은 결과가 나왔다. Best hyperparameters인 경우는 'max\_depth': 4, 'max\_features': 'log2', 'n\_estimators': 376인 경우로 나온 것이다. 100번의 서로 다른 파라미터를 가질 때의 학습 결과를 가지고 그중 가장 최적의 파라미터를 찾아 낸 것이다.

실습에 비하여 Validation ACC값이 높아지고 Train ACC값과의 차이가 높아졌으므로 일반화가 더 잘 되었고 더 좋은 성능을 낸다는 것을 확인할 수 있다.