HW5\_20210815\_정재현

A. classification

먼저 Ensemble을 이용하여 classification을 할 때, RandomForest 기법을 사용하기 위해서 RandomForestClassifier를 import 해주었다. 우리에게 주어진 파일들을 분석해보면 train\_classification.csv와 test\_classification.csv가 존재한다. 이때 train\_classification.csv를 보면 Sector와 Soil이 문자형으로 되어 있는 것을 확인할 수 있었다. 그래서 이를 int64형태로 바꾸기 위해 인코딩을 진행하였다. 먼저 Sector의 경우 값이 Sector\_정수 형태로 되어 있어 문자열을 7번째 자리로 잘라서 int 함수를 사용하여 값을 정수 값으로 바꿔주고, Soil 같은 경우 Soil\_정수 값으로 나타나져 있어서 5번째 값부터 받아 int 함수를 사용하여 int 64형태로 만들어주었다. 그 이후 astype을 이용하여 Sector와 Soil행에 대해 int64형으로 인코딩을 시켜주었다.

그 이후 데이터 라벨을 분리 했는데 Plant\_species라는 행을 추출하여 train\_label을 만들어 주었고, 그 이후 Plant\_species라는 행을 드롭한 train\_class를 만들어 주었다. 그 이후 RandomForest classification을 할 텐데 주석 처리한 부분은 random\_state값을 0부터 100까지 돌리면서 가장 최적의 random\_state값을 찾는 부분을 주석 처리 하였다. 이 부분은 train\_class를 분리하여 train과 test를 직접 알게 한 후 가장 최적의 random\_state값의 predict하는 것을 찾는 것이다. 그래서 plot을 이용하여 validation Accuracy를 구해볼 수 있었다. 이 결과 x(=random\_state)=37일 때 가장 정확도가 높다는 것을 알 수 있었고 이를 통해 RandomForest 패키지에서 재공하는 RandomForestClassifier 함수를 사용하여 random\_state=37을 부여해주고 그리고 fit을 사용하여 train\_class와 train\_label을 넣어주고 test\_class(test\_classification.csv)를 predict 한 값을 Category라는 리스트에 저장을 시키도록 하였다.

이후 값을 csv로 만들기 위해 dataframe을 구성하였는데, Id라는 리스트에 range2000을 사용하여 0부터 19999까지의 값을 담고 dataframe에 Id와 아까 추출한 Category를 행으로 하고 값을 각각 넣어준 후 to\_csv로 sample\_classification.csv로 저장하는데, 이때 샘플 값으로 주어진 파일에서는 index가 없었으므로 index=False라는 서식을 부여하여 정상적으로 RandomForest를 활용한 Ensemble classification을 완료하였다.

이외에도 randomforest의 하이퍼파라미터를 변경시켜보면서 n\_estimator 등을 조정해보았으나 위의 방법이 내가할 수 있는 최적의 결과를 가져다주었다.

B. Regression

마찬가지로 Ensemble의 RandomForest를 이용한 regression을 사용하였다. RandomForestRegressor을 import하였다. 우리에게 주어진 파일인 train\_regression.csv와 test\_regression.csv 파일이 있는데 이것을 행을 분석해보니 모두 float64형이라 인코딩이 따로 필요하지는 않았다. 그래서 바로 라벨을 추출하였는데 여기서는 score라는 행을 추출하여 train\_label이라는 dataset을 만들고, 그리고 score을 drop한 train\_reg를 읽어왔다.

그 이후 주석 처리한 부분이 있는데 이 부분 역시 RandomForest에서 0부터 100의 random\_state 값을 대입하여 가장 적합한 random\_state 값을 찾는 과정 이었다. RandomForestRegressor을 이용하여 random\_state값을 변화시키며 predict한 값을 찾는 것으로 mean\_squared\_error나 math 라이브러리를 사용하여 가장 적합한 random\_state값을 찾아보았는데, 이결과 random\_state = 91일 때로 RandomForestRegressor에 random\_state = 91을 넣고 fit을 사용하여 train\_reg와 train\_label로 학습 시키고 test\_reg(test\_regression.csv)을 predict한 결과를. Predicted라는 리스트에 저장을 해두었다. 마찬가지로 A에서 classification 결과 처럼 Id라는 리스트에 range2000을 사용하여0부터 1999의 값을 가지도록 하고 dataframe을 만들어 Id와 Predicted라는 행을 해당하는 리스트와 연결 시켜 만들었고, index=False를 사용하여 sample\_regression.csv파일로 저장하도록 to\_csv를 통해 RandomForestRegressor을 완료하였다.

이때도 n\_estimator는 100으로 하였고 마찬가지로 다른 하이퍼 파라미터를 조정했을 때보다 위의 방법이 가장 정확도가 높아서 해당 풀이방법대로 해결하였다.