20210815\_정재현

먼저 knn classification을 사용하기 위해서 KNeighborsClassifier를 import 해주었다. 우리에게 주어진 파일들을 분석해보면 라벨링 된 데이터와 아닌데이터가 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 Knn classification을 할 텐데 주석 처리한 부분은 k값을 1부터 100까지 돌리면서 가장 최적의 k값을 찾는 부분을 주석 처리 하였다. 이 부분은 train\_class를 분리하여 train과 test를 직접 알게 한 후 가장 최적의 k값의 predict하는 것을 찾는 것이다. 그래서 plot을 이용하여 validation Accuracy를 구해볼 수 있었다. 이 결과 K=1일 때 가장 정확도가 높다는 것을 알 수 있었고 이를 통해 knn 패키지에서 재공하는 KNeighborsClassifier 함수를 사용하여 k=1을 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 knn classification을 완료하였다.