**Assignment 2: Comparison of Supervised Learning Algorithms**

2016314726 정영준

손글씨 데이터셋을 sklearn의 load\_digits을 사용하여 불러왔습니다. Data, image, target, target\_names 총 4개의 부분으로 데이터셋은 나뉘어 있었으며 data는 shape를 사용하여 확인한 결과 1797개의 row, 64개의 column으로 이루어져 있었습니다. image에는 같은 1797개의 데이터 지만 3차원 데이터로 feature가 8x8로 표현되어 있었습니다. 이를 matplotlib.pyplot의 imshow를 통해 확인해본 결과 실제 8x8개의 cell로 이루어진 색깔이 없는 손글씨 이미지였습니다. target값을 확인해본 결과 0 ~ 9로 각 숫자에 맞춰서 target값이 데이터에 알맞게 주어진 것을 알 수 있었습니다. 먼저 1797개의 데이터를 train set, test set 2개로 나누었습니다. Train\_test\_split의 stratify를 이용하여 target이 골고루 train과 test 세트에 분포하도록 하였고 test set의 사이즈는 전체의 20%로 설정하였습니다. 크기가 큰 이미지 데이터가 아니기 때문에 전처리는 간단하게 StandardScaler, MinMaxScaler 2개를 사용하였습니다. K-nearest Neighbors, Logistic Regression, Decision Tree, Random Forests, Support Vector Machine, 총 5개의 모델을 각 parameter를 조정하고, 다른 전처리를 적용하며 train set 내에서 5개의 fold로 나누어 교차검증을 수행하였습니다. Sklearn의 Pipeline을 활용하여 K-nearest Neighbors 모델에서는 n\_neighbors parameter를 3,5,7,9,11로 변경하였습니다. Decision Tree, Random Forest에서는 max\_depth, min\_samples\_leaf parameter를 바꿔가면서 학습을 시켰으며, Logistic Regression에서는 multi\_class를 ovr로 설정하여 target의 종류가 10개인 데이터를 분류할 수 있도록 하였고 max\_iter 값을 default 값보다 크게 설정하고, C값을 조정하며 학습시켰습니다. Decision Tree와 이를 weak estimator로 활용하는 Random Forest에서는 Scaler를 사용해도 결과가 변하지 않기 때문에 Scaler를 사용하지 않았습니다. 나머지 모델들 에서는 MinMaxSclaer, StandardScaler, 전처리 하지 않음 3개의 경우에서 모두 퍼포먼스를 측정하였습니다. 각 5개의 fold에 대해 valid score를 accuracy로 측정하고 이를 평균을 낸 mean\_test\_score를 hyperparameter 튜닝을 위한 평가기준으로 삼았습니다. 0.989로 가장 높은 점수가 나온 모델은 MinMaxSclaer(최소 0, 최대1)을 적용하고 hyperparameter를 C=100, gamma=0.1로 설정한 SVC였습니다. 다음으로 MinMaxScaler를 사용하거나 또는 전처리를 하지않은 n\_neighbors=3으로 설정된 K-nearest Noegibors 모델이었습니다. 0.985의 정확도를 기록하였습니다. 가장 점수가 낮았던 모델들은 scaling이 되지 않은 SVC모델이었습니다. 10%의 정확도를 기록하였으며 이는 한 레이블만 선택하여도 얻을 수 있는 정확도로 모델이 거의 제대로 학습하지 못했다는 것을 알 수 있습니다. StandardScaler를 적용한 SVC모델도 하위권의 점수를 기록하였습니다.

최적의 Learning Algorithm과 Hyperparameter 설정은 MinMaxScaler로 전처리가 된 C=100, gamma=0.1로 설정된 SVC 모델이었습니다. Test set에서 정확도를 측정한 결과 99.17% 로 다른 모델들 보다 뛰어난 성능을 보였습니다.







