AI 레포트

> [19반] 정민석_7000

AI 모델 성능 측정(기본)과 AI 모델 최적화(심화)를 수행하였습니다.

AI 모델 성능 측정(기본)

성능 측정 결과

성능 측정의 지표는 정확도, 정밀도, 재현율, F1 스코어 등이 있습니다. 정확도는 모델이 정확히 분류한 샘플의 비율이고, 정밀도는 양성으로 예측한 생플 중 실제로 양성인 샘플의 비율이며, 재현율은 실제로 양성인 샘플 중 양성으로 예측한 샘플의 비율입니다. F1 스코어는 정밀도와 재현율의 조화평균입니다. 그리고 대체로 정확도는 데이터가 균형적인 경우, F1 스코어는 데이터가 불균형한 경우 사용합니다.

지금부터는 실습 4에서의 모델들을 분류하고, 성능측정해본 결과를 소개하겠습니다.

단일 모델

단일 모델은 하나의 결정트리로 출력값을 예측하는 모델입니다.

- 의사결정 트리

의사결정 트리는 주어진 입력값들의 조합에 기반한 규칙을 바탕으로 출력값을 예측하는 모델입니다. 예를 들어 Height와 Weight라는 라벨이 있을때, Height가 170을 기준으로 크냐 작냐로 나누고, Weight가 60을 기준으로 나누는 방법으로 출력값을 예측합니다. 다음은 실습 4에서 진행한 의사결정 트리 모델의 정확도를 측정한 것입니다.

```
[[1307   63]
 [  73  406]]
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      1370
           1       0.87      0.85      0.86       479

    accuracy                           0.93      1849
   macro avg       0.91      0.90      0.90      1849
weighted avg       0.93      0.93      0.93      1849

F1 Score :  0.8565400843881856
```

앙상블 모델

앙상블 모델은 여러개의 결정트리를 이용하여 출력값을 예측하는 모델입니다. 배깅, 부스팅, 스태킹의 방법을 이용할 수 있습니다.

- 랜덤 포레스트

랜덤 포레스트는 배깅 방식의 대표적인 모델 중 하나입니다. 데이터를 랜덤으로 추출하고, 합하여 학습을 진행합니다. 아래는 실습 4에서 진행한 랜덤 포레스트 모델의 성능 측정 결과입니다.

```
[[1356   14]
 [  67  412]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      1370
           1       0.97      0.86      0.91       479

    accuracy                           0.96      1849
   macro avg       0.96      0.92      0.94      1849
weighted avg       0.96      0.96      0.96      1849

F1 Score :  0.9104972375690608
```

정확도를 높일 수 있는 방법

가장 간단하며 효과적인 방법은 적절한 데이터의 개수를 늘리는 것입니다. 지금까지 실습에서 학습한 AI 모델은 많은 데이터를 기반으로 통계적으로 확률이 높은 결과를 예측하는 것입니다. 즉, 통계적으로 확률이 높게하기 위해서는 많은 수의 표본을 기반으로 예측하는 것이 필요합니다. 따라서 학습 데이터의 개수를 폭발적으로 늘리는 방향이 있겠습니다.

더하여 모델을 학습할 때에, 데이터의 순서를 랜덤하게 배치하는 것도 중요하다고 생각합니다. 실습에서 학습한 AI 모델은 결국 예측치와의 loss를 줄이는 방향으로 학습을 진행합니다. 이때, 학습 데이터의 순서 자체를 학습하여, 순서를 기반으로 loss를 줄이는 잘못된 학습을 진행하는 경우가 있는 것으로 압니다. 따라서 학습 데이터 자체의 순서를 랜덤하게 배치하여, AI가 적절한 방식으로 학습하도록 유도하는 것이 중요하겠습니다.

AI 모델 최적화(심화)

사용전략

Caviar 전략을 활용하여 모델을 생성했습니다. Caviar 전략은 여러 모델을 생성하고, 최적의 모델을 선택하는 전략입니다.

전/후 정확도 비교

```
최적화 전 F1 Score :  0.9110867178924259
최적의 파라미터 값 :  {'max_depth': 20, 'n_estimators': 500}
최고의 점수 :  0.9131979282390897
최적화 후 F1 Score :  0.9094922737306843
```

더하여 F1 점수는 오히려 하락한 수치를 보였습니다. 최적화 전에는 약 0.911이었으나, 최적화 이후에는 0.909로 낮은 수치를 보였습니다. 하지만 Caviar 전략의 목적 중 하나는 모델의 경량화가 포함됨으로, F1 수치와 모델의 경량화 사이에서의 균형을 찾았다고 생각합니다.

하이퍼 파라미터 수치

아래와 같이 더 많은 결정 트리를 만드는 것으로 파라미터 수치를 튜닝하였습니다.

```
# 기존
[ {'n_estimators':[100,200,300,400,500], 'max_depth':[10,20,30,40,50]}]
```

```
# 변경
[ {'n_estimators':[500,600,700,800,900], 'max_depth':[10,20,30,40,50]}]
```

그러나 전 후 모두 n_estimators: 500, max_depth: 20에서 가장 좋은 결과가 도출되었습니다.

결과

F1 점수는 약 0.911에서 0.909으로 하락하였으나, 모델의 경량화로 인하여 수치가 낮아졌다고 생각합니다. 하이퍼 파라미터의 경우, 더 많은 결정트리를 만드는 방향으로 튜닝하였으나, n_estimators: 500, max_depth: 20에서 가장 좋은 결과를 보였습니다.

한계점

하이퍼파라미터를 통하여 튜닝을 할때, 너무 많은 결정트리를 만드는 것은 모델의 성능에 큰 도움을 주지 못하는 것처럼 보입니다. 많은 컴퓨팅 파워를 동원하여, 특이점 이상으로 학습 데이터를 늘리거나, 모델 자체의 로직을 최적화하는 등의 근본적인 노력이 필요해 보입니다.