학습자 데이터 머신러닝 프로젝트

빅데이터 8기 이현희

사용된 머신러닝 모델

DecisionTreeRegressor

선정 이유

1. 직관적이고 이해하기 쉬운 모델 : 트리 구조로 시각화할 수 있어 데이터가 어떻게 예측되는지를 직관적 이해 가능

2. 복잡한 비선형 관계를 모델링 가능 : 데이터의 비선형적인 관계를 잘 포착하여 다양한 패턴을 학습하는 데 유리

사용된 머신러닝 모델

DecisionTreeRegressor 결과

dt.score(sub_input, sub_target) : 1.0
dt.score(val_input, val_target) : 0.6897421524663677

머신러닝 모델의
하이퍼파라미터 튜닝을 위한 방법

cross_validate 결과

```
np.mean(scores['train_score']) : 1.0
np.mean(scores['test_score']) : 0.7414658644988156
```

머신러닝 모델의
하이퍼파라미터 튜닝을 위한 방법

GridSearch

선정 이유

1. 최적의 하이퍼파라미터 찾기: 모델의 성능을 극대화하기 위해 다양한 하이퍼파라미터 조합을 시험해볼 수 있습니다.

2. 교차 검증을 통한 신뢰성 확보: 교차 검증(cross-validation)을 통해 각 조합에 대한 모델 성능을 평가하므로,
과적합을 방지하고 일반화 성능을 향상시킬 수 있습니다.

머신러닝 모델의
하이퍼파라미터 튜닝을 위한 방법

GridSearch

- **'max_depth':** 트리의 최대 깊이

- **'min_impurity_decrease':** 노드를 분할하기 위한 최소 샘플 수

- **'min_samples_leaf':** 리프 노드에 필요한 최소 샘플 수

- **'min_samples_split':** 분할을 위한 최소 불순도 감소량

머신러닝 모델의
하이퍼파라미터 튜닝을 위한 방법

GridSearch

{'max_depth': 20,
'min_impurity_decrease': np.float64(0.0001),
'min_samples_leaf': 8,
'min_samples_split': 2}

dt.score(sub_input, sub_target) : 0.9097578126547302
dt.score(val_input, val_target) : 0.8598681472263008

머신러닝 모델의
하이퍼파라미터 튜닝을 위한 방법

cross_validate 결과

Average train score: 0.9069794448550719
Average validation score: 0.853042221428782

Result

test 데이터를 이용한 결과

```
print(dt.score(test_input, test_target))
```

```
0.8258047179780313
```

test prediction과
actual values 비교

```
print("Test predictions:", test_predictions)
print("Actual values:", test_target)
```

```
Test predictions: [ 8.22222222 11.9        18.07692308 10.4         3.5        12.75
 13.         10.92857143  8.22222222 13.5         9.625      13.5
  9.125      14.09090909  0.          3.5        15.2        12.75
  8.22222222 15.2        15.71428571 10.4        10.92857143 10.4
 15.71428571  7.33333333 10.4        11.9        11.9        12.75
 15.2         0.          7.33333333 10.92857143 18.07692308 13.5
 18.07692308 12.75       18.07692308 13.         11.9        13.
 13.         14.09090909 14.09090909 11.9        15.71428571  9.625
  5.75        8.22222222  0.          9.625       8.875      13.5
  0.          3.5        14.09090909 10.4         3.5         8.22222222
 14.09090909 15.2        13.         10.92857143  9.125       8.22222222
  5.75       10.4        12.75       11.9        13.         13.
  8.22222222 10.4         6.33333333  8.875       9.125      15.71428571
 10.92857143 10.92857143]
Actual values: [ 8 11 19 10  9 11 14 12 11 13  8 15 10 14  0 10 14 11  8 15 16 10 11  9
 18  8  9 11 11 11 15  0  9 11 19 12 18 13 18 13 11 13 13 14 14 11 15  9
  5 10  0  9  8 13  0 11 14 10  0  9 14 15 13 12 10  8  6 11 12 12 12 14
 10 10 10 10 10 15 11 11]
```