기계학습 3주차 과제

자동차 가격 예측

컴퓨터공학과
17011584 정재경

개요

주어진 데이터셋은 중고 자동차에 대한 정보가 포함돼있어, 각 정보에 따른 중고차 가격을 나타낸다.
제공된 정보에 따라서 중고 자동차의 가격을 예측하는 문제로, 회귀문제에 해당한다.
이번 과제는 K-Nearest Neighbors 방식을 사용해서 해결해보았다.

변경 가능한 하이퍼파라미터

scikit learn 에서 제공하는 KNeighborsRegressor 공식문서에 따르면 커스터마이징이 가능한 하이퍼 파라미터에는 n_neighbors, weights, algorithm, leaf_size, p, metric, metric_params, n_jobs로 총 7가지이다.
주요한 하이퍼 파라미터의 의미를 살펴보면

1. n_neighbors - KNN알고리즘에서의 K값을 의미한다. 일반적으로 가장 가까운 것들을 투표의 형식으로 뽑은 것이기 때문에 짝수를 사용하면 동점이 나오는 경우가 생길 수 있기 때문에 홀수를 사용하는 것이 좋다
2. weights - 거리를 처리하는 방식이다. 가능한 값들은 uniform과 distance 가 있다. uniform은 거리에 상관없이 모두 같은 영향력을 행사하는 것이고, distance의 경우는 거리가 가까울 수록 더 높은 점수를 주는 방식이다. default 값은 uniform이다.
3. algorithm - 거리를 계산할 때 사용할 알고리즘이다. default 값은 auto로, 들어온 값에 따라서 최적의 알고리즘을 선택한다.
4. leaf_size - 거리계산 알고리즘에 사용될 파라미터 값이다.
5. p - Minkowski 공간에서 사용될 p 값이다. p가 1일 때는 L1거리 (Manhattan Distance) 2일 때는 L2거리 (Euclidean distance)와 같다.

실험

우선, 이 과제는 테스트 데이터셋의 outlier들이 몇몇 존재하는 것을 알 수 있었다. year이 2060으로 들어간 row, transmission이 유일하게 2인 row, fueltype이 1인 row가 단 두개여서 outlier이라고 판단했다. 따라서 이 4개를 제거한 데이터셋으로 학습을 진행했다.

이번 과제에서 변경을 해보며 결과값의 차이 추이를 본 하이퍼파라미터는 n_neighbors, weights 이다.
다음과 같은 함수를 작성하여 n값을 번경해가며 모델을 학습시키고, 학습 데이터셋과 테스트 데이터셋의 MAE 점수를 계산하여 result 배열에 저장했다.

```
def _predict(n, train_set, train_ans, result:list, weights):
    knn = KNeighborsRegressor(n_neighbors=n, weights=weights)
    knn.fit(train_set, train_ans)
    train_pred = knn.predict(train_set)

    current = {
        'n': n,
        'weights': weights,
        'Train': mean_absolute_error(train_ans, train_pred),
        }
    result.append(current)
```

[5] ✓ 0.1s Python

이렇게 계산된 결과를 i값을 3부터 17까지 넣어서 점수를 기준으로 정렬해본 결과, n=7일 때, weights=distance 일 때가 가장 높은 점수 (낮은 MAE값) 가 나왔다.

```
result = []
for i in range (3, 19, 2):
    _predict(i, x, y, result, 'uniform')
for i in range (3, 19, 2):
    _predict(i, x, y, result, 'distance')

pd.DataFrame(result).sort_values(by=['Train'], ascending=True)
```

[8] ✓ 33.4s

	n	weights	Train
10	7	distance	19.970358
13	13	distance	19.977352
14	15	distance	19.977352
15	17	distance	19.977352
12	11	distance	19.980396
11	9	distance	19.984793
9	5	distance	19.987600
8	3	distance	20.141603
0	3	uniform	935.369032
1	5	uniform	1093.089130
2	7	uniform	1186.507125
3	9	uniform	1252.925875
4	11	uniform	1306.834493

결론

따라서 이 데이터셋은 N=7일 때, weights=distance일 때가 가장 성능이 좋을 것이라 예측하였다. 눈 여겨볼 만한 점은 uniform일 때와 distance일때의 오차값 차이가 상당히 크다는 점이다. 이는 똑같이 KNN방식으로 분류문제를 풀 때는 큰 영향이 없었는데, 회귀문제를 풀어보니 차이가 크게 나타났다.
조금 더 정밀하게 값을 예측해야하는 회귀문제가, 어떤 분류 안에만 들어가는지만 예측하면 되는 분류문제보다 값을 판단함에 있어서 거리를 중요하게 생각해야하는 정도가 다른 것으로 판단된다.

실제 Kaggle에 제출할 때도 N=7, weights=distance으로 제출한 점수가 가장 높았다.

노트북 링크: https://github.com/therealjamesjung/ML_2022/blob/master/Assignment%204/Assignment-4.ipynb