Dacon_nuclear_study_1st_3조

3 조 2 주차 발표자료_데이터전처리

데이터 전처리

데이터 전처리의 필요성

데이터에 Null값 등 결손값은 허용되지 않는다!

➔ Null값을 고정된 다른 값으로 변환해야 함

어떻게 결손값을 처리할 것인가?

1) 결손값이 단순 결손인 경우(결손값이 적은 경우!) : 피처의 평균값 등으로 간단히 대체

2) 결손값이 많은 경우 : 해당 피처는 드롭하는 것이 나을 수도 있음

만약, 피처의 중요도가 높다면 Null을 단순히 피처의 평균값으로 대체할 경우 예측 왜곡이 심할 수 있음

➔ 정밀한 대체 값을 선정해야 함!

사이킷런의 머신러닝 알고리즘은 문자열 값을 입력 값으로 허용하지 않음

➔ 따라서 모든 문자열 값은 인코딩 해서 숫자 형으로 변환하기!

데이터 인코딩

데이터 인코딩 방식 : 레이블 인코딩과 원-핫인코딩

레이블 인코딩

카테고리 피처를 코드형 숫자 값으로 변환하는 것

즉, 간단하게 문자열 값을 숫자형 카테고리 값으로 변환

사이킷런의 LabelEncoder 클래스로 구현

숫자 값의 크기에 대한 특성 작용 -〉 선형회귀알고리즘에는 적용하지 말 것

단 트리 계열의 ML알고리즘은 숫자의 특성 반영하지 않으므로 적용 가능

```
#2020-03-24
#jih020202@gmail.com
from sklearn.preprocessing import LabelEncoder
```

```
items=['월요일','일요일', '금요일','금요일','목요일', '화요일', '수요일', '토요일' ]
```

```
#LabelEncoder 객체 생성 후 레이블 인코딩

encoder = LabelEncoder()
encoder.fit(items)
labels = encoder.transform(items)
print('인코딩 : ',labels )
```

```
인코딩 :  [3 4 0 0 1 6 2 5]
```

```
#어떻게 인코딩 되었는지 속성값 확인 가능

print('인코딩 클래스 : ', encoder.classes_)
```

```
인코딩 클래스 :  ['금요일' '목요일' '수요일' '월요일' '일요일' '토요일' '화요일']
```

```
#디코딩

print('디코딩 : ', encoder.inverse_transform([0,1,2,3,4,5,6]))
```

```
디코딩 :  ['금요일' '목요일' '수요일' '월요일' '일요일' '토요일' '화요일']
```

〈예시〉

원-핫 인코딩

피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표시하고 나머지 칼럼에는 0을 표시하는 방식

사이킷런의 OneHotEncoder 클래스로 구현

➔ 이때 변환 전 모든 문자열 값이 숫자형 값으로 변환되어야 함 + 입력 값으로 2차원 데이터 필요

```
#2020-03-24
#jih020202@gmail.com
from sklearn.preprocessing import OneHotEncoder
import numpy as np
```

```
items=['월요일','일요일', '금요일','금요일','목요일', '화요일', '수요일', '토요일' ]
```

```
#숫자 형태로 변환
encoder = LabelEncoder()
encoder.fit(items)
labels = encoder.transform(items)

#2차원으로 변환
labels = labels.reshape(-1,1)
```

```
#원핫인코딩 적용
encoder = OneHotEncoder()
encoder.fit(labels)
oh_labels=encoder.transform(labels)
print('인코딩 데이터 : ₩n', oh_labels.toarray())
print('데이터 차원 :₩n', oh_labels.shape)
```

```
인코딩 데이터 :
 [[0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0.]]
데이터 차원 :
 (8, 7)
```

〈예시〉

판다스의 get_dummies()로 구현 가능

➔ 문자열 카테고리 값을 숫자 형으로 변환할 필요 없음

```
In [27]: #2020-03-24
         #jih0202020@gmail.com
         import pandas as pd

In [28]: df = pd.DataFrame({'item' : ['월요일','일요일', '금요일','금요일','목요일', '화요일',

In [29]: pd.get_dummies(df)
```

Out[29]:

	item_금요일	item_목요일	item_수요일	item_월요일	item_일요일	item_토요일	item_화요일
0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0
2	1	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0
5	0	0	0	0	0	0	1
6	0	0	1	0	0	0	0
7	0	0	0	0	0	1	0

〈예시〉