머신러닝 프로젝트 처음부터 끝까지

202155603 정윤서

텍스트와 범주형 특성 다루기

- 사이킷런의 OridinalEncoder 클래스 사용

```
from sklearn.preprocessing import OrdinalEncoder
ordinal_encoder = OrdinalEncoder()
housing_cat_encoded = ordinal_encoder.fit_transform(housing_cat)
housing_cat_encoded[:10]
```

```
array([[0.],
       [0.],
       [4.],
       [1.],
       [0.],
       [1.],
       [0.],
       [1.],
       [0.],
       [0.]])
```

머신러닝 알고리즘이 가까이 있는 두 값이 떨어져 있는 두 값보다 더 비슷하다고 생각한다는 점이 문제

→ 일반적으로 카테고리별 이진 특성을 만들어 해결

(원-핫 인코딩one-hot encoding)

텍스트와 범주형 특성 다루기

- 사이킷런의 OneHotEncoder 클래스 사용

```
from sklearn.preprocessing import OneHotEncoder
cat_encoder = OneHotEncoder()
housing_cat_1hot = cat_encoder.fit_transform(housing_cat)
```

```
housing_cat_1hot<16512x5 sparse matrix of type '<class 'numpy.float64'>'
with 16512 stored elements in Compressed Sparse Row format>
```

출력을 보면 넘파이 배열이 아니고 사이파이SciPy 희소 행렬sparse matrix

- 수천 개의 카테고리가 있는 범주형 특성일 경우 매우 효율적
- 0을 모두 메모리에 저장하는 것은 낭비이므로 희소 행렬은 0이 아닌 원소의 위치만 저장

범주형 변수 인코딩

id	bin_0	bin_1	bin_2	bin_3	bin_4	nom_0	nom_1	nom_2	nom_3	nom_4	...	nom_9	ord_0	ord_1	ord_2	ord_3	ord_4	ord_5	day	month	target
0	0	0	0	T	Y	Green	Triangle	Snake	Finland	Bassoon	...	2f4cb3d51	2	Grandmaster	Cold	h	D	kr	2	2	0
1	0	1	0	T	Y	Green	Trapezoid	Hamster	Russia	Piano	...	f83c56c21	1	Grandmaster	Hot	a	A	bF	7	8	0
2	0	0	0	F	Y	Blue	Trapezoid	Lion	Russia	Theremin	...	ae6800dd0	1	Expert	Lava Hot	h	R	Jc	7	2	0
3	0	1	0	F	Y	Red	Trapezoid	Snake	Canada	Oboe	...	8270f0d71	1	Grandmaster	Boiling Hot	i	D	kW	2	1	1
4	0	0	0	F	N	Red	Trapezoid	Lion	Canada	Oboe	...	b164b72a7	1	Grandmaster	Freezing	a	R	qP	7	8	0

5 rows × 24 columns

id	bin_0	bin_1	bin_2	bin_3	bin_4	nom_0	nom_1	nom_2	nom_3	nom_4	...	nom_8	nom_9	ord_0	ord_1	ord_2	ord_3	ord_4	ord_5	day	month
300000	0	0	1	T	Y	Blue	Triangle	Axolotl	Finland	Piano	...	9d117320c	3c49b42b8	2	Novice	Warm	j	P	be	5	11
300001	0	0	0	T	N	Red	Square	Lion	Canada	Piano	...	46ae3059c	285771075	1	Master	Lava Hot	I	A	RP	7	5
300002	1	0	1	F	Y	Blue	Square	Dog	China	Piano	...	b759e21f0	6f323c53f	2	Expert	Freezing	a	G	tP	1	12
300003	0	0	1	T	Y	Red	Star	Cat	China	Piano	...	0b6ec68ff	b5de3dcc4	1	Contributor	Lava Hot	b	Q	ke	2	3
300004	0	1	1	F	N	Red	Trapezoid	Dog	China	Piano	...	f91f3b1ee	967cfa9c9	3	Grandmaster	Lava Hot	I	W	qK	4	11

5 rows × 23 columns

범주형 변수 인코딩

변수명	bin_*	ord_*	nom_*	day, month
의미	이진 변수	순서 변수	명목 변수	날짜 변수
인코딩 방법	굳이 할 필요 X	순서를 보존하여 숫자 형태로 인코딩	원-핫 인코딩	원-핫 인코딩

범주형 변수 인코딩

- 이진 변수(bin_*) 인코딩
  - 딱히 인코딩 해줄 필요 없다.
  - 0과 1로 구성되어 있기 때문에
- T, F나 Y, N처럼 문자로 되어 있는 경우
  - 각각 1과 0으로 변환

범주형 변수 인코딩

- 순서 변수(ord_*) 인코딩
  - 순서(순위)가 있는 데이터
  - 순서(순위)를 보존해야 한다.
- 숫자 간의 간격이 너무 차이는 경우
  - 스케일링 해준다.
  - ex) 사이킷런에서 제공하는 표준화 스케일러(StandardScaler) 사용

범주형 변수 인코딩

- 명목 변수(norm_*) & 날짜 변수(day, month) 인코딩
  - 원-핫 인코딩(One-Hot Encoding)을 주로 사용

https://suy379.tistory.com/157

범주형 변수 인코딩

가중치 발생

- Label Encoding
  - ex) ‘English’, ‘Korean’, ‘Math’ 를 0, 1, 2 처럼 변환하는 것

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder() df_c['subject'] =
encoder.fit_transform(df_c['subject'].values)
```

  - 인코딩 값에 대한 원본값을 알고 싶을때에는 .classes_
  - 인코딩 되기 전으로 되돌리고 싶을 때에는 inverse_transform() 을 사용

```
display(encoder.classes_)
display(encoder.inverse_transform(df_c['subject']))
```

변주형 변수 인코딩

- One-Hot Encoding
  - Label Encoding 이 선행되어야 함
- get_dummies()
  - LabelEncoder() + OneHotEncoder() = get_dummies()
  - 두 과정을 한 번에 모두 처리해주는 함수

```
pd_df = pd.get_dummies(df['subject']) df_result =
pd.concat([df, pd_df], axis=1)
```

	subject	score	english	korean	math	science
0	math	50	0	0	1	0
1	english	40	1	0	0	0
2	science	60	0	0	0	1
3	math	80	0	0	1	0
4	korean	90	0	1	0	0
5	science	10	0	0	0	1

https://hwi-doc.tistory.com/entry/%EB%B2%94%EC%A3%BC%ED%98%95-%EB%B3%80%EC%88%98-%EC%B2%98%EB%A6%AC-%EB%B0%A9%EB%B2%95