Categorical Feature Encoding project

MACHINE LEARNING

목차

20212543 김서령

20160393 황건하

20212546 김예향

항목1

01. Categorical Feature Encoding 기법

Encoder	원 리	특 징
One Hot Encoding	각 범주형 변수의 값들을 1과 0으로 매핑시키는 것	모델링의 학습속도가 매우 느림 일반적으로 낮은 low-cardinality 범주형 변수에 많이 사용
Ordinal Encoding	레이블 인코딩과 비슷하지만, 이 기법은 범주형 변수 중 순서 뿐만 아니라 알파벳순서까지도 고려하는 방식	순서형 자료에 매우 적합함 선형회귀(예측)에 적절하지 않음
Binary Encoding	범주형 변수값들을 이진법으로 변환해주는 방법	one-hot encoding에 비해 훨씬 더 적은 개수의 추가 변수가 생성돼서 모델링 학습시 속도가 더 빠름
Frequency Encoding	범주형 변수 값들의 빈도수에 기반해 인코딩하는 방식	범주형 변수의 빈도값이 종속변수와 연관이 있게 된다면 모델이 가중치를 부여하도록 도와줌
Target Encoding	각 범주별 target variable의 평균으로 label encoding 하는 방법	빠르게 학습하지만 과적합 문제에 매우 취약 레이블 인코딩과 비슷하지만, 범주형태의 독립변수들이 종속변수의 값들과 상관관계가 있음을 가정하는 것이 차이점
Weight of Evidence Encoding	카드사기와 같이 2개의 결과인 경우 범주별로 log odds를 구해 label encoding 하는 방법	이분법에 적합하기 때문에 로지스틱 회귀에 매우 자주 사용됨 정보 손실량이 발생하는 단점이 있음
Leave One Out Encoding	mean encoding과 동일하나 평균을 구할때, 해당 row의 값을 제외하고 구해 encoding 하는 방법	이상치를 제외시킨 상태에서의 종속변수값들의 평균을 이용해 범주형 변수값들을 인코딩하는 방법
James-Stein Encoding	관측된 feature value와 관측되지 않은 feature value의 종속변수 값의 평균값들 중에서 다시 가중치를 부여한 평균값을 이용해 인코딩하는 방법	원본 데이터들의 분포가 정규분포일 때만 적용이 유의미함
M-Estimate Encoding	Mean encoding을 단순화시킨 방법	m값은 정규화의 가도를 의미하며 m값을 높게 부여할수록 더 강력하게 정규화시키면서 제한강도가 높아짐
CatBoost Encoding	현재 데이터의 인코딩을 위해 이전 데이터들의 인코딩 방법	각 모형마다 다른 데이터를 사용했기 때문에, target leakage문제가 사라짐 Sparse 한 Matrix 는 처리하지 못함

02. Categorical Feature Encoding 전략

01 결측값 처리는 최빈값으로 대체

02 모델 학습 시간이 너무 많이 소요되기에 처음부터 X_train과 y_train 데이터를 1/5 정도 샘플링한 후 학습 진행

`(600000, 24)` ⟫ `(120000, 24)`

03 DecisionTree 하이퍼파라미터를 max_depth = 3과 random_state = 42로 고정

Logistic Regression와 MLP 하이퍼파라미터를 random_state = 42로 고정

04 데이터를 살펴보니, cardinality가 높다는 것을 알게 되었고, 사전에 조사한 인코더 특징에 따라 cardinality가 높을 때 유용한 베이지안 기반의 인코더 기법에 초점을 맞춰 실험 진행

05 베이지안 기반의 인코더 기법에는 Target, James-Stein, Weight of Evidence, M-estimator, CatBoost 인코딩이 있음

06 기존의 Target Encoding은 오버피팅과 데이터 손실 문제를 일으키므로, 이에 대한 해결책으로 현재 타켓값을 사용하지 않고, 이전 데이터들의 타켓 값만을 사용하는 CatBoost Encoding 기법을 추가적으로 사용하여 실험 진행

07 CatBoost Encoding은 오버피팅과 데이터 손실 문제를 막아주기 때문에 인코딩 전략으로 수립

항목1

03. 모델별 인코딩 성능 비교

모델별로 어떤 Categorical Feature Encoding 기법이 유용한가?

Encoder	Logistic	Decision Tree	MLP
One-hot	0.76706	0.60483	0.6971
Ordinal	0.63992	0.65438	0.63633
Binary	0.68773	0.62133	0.66638
Frequency	0.6234	0.64538	0.66021
Target	0.761	0.6398	0.76166
Weight of Evidence	0.76067	0.63964	0.75208
Leave One Out	0.77574	0.5	0.77519
James-Stein	0.75608	0.6376	0.75609
M-Estimator	0.76177	0.63981	0.76193
CatBoost	0.77015	0.65618	0.77321

성능 순위	Logistic	Decision Tree	MLP
1순위	leave one out	catboost	leave one out
2순위	catboost	ordinal	catboost
3순위	one-hot	Frequency	m-estimator
4순위	m-estimator	m-estimator	target
5순위	target	target	james-stein
6순위	weigh of evidence	weigh of evidence	weigh of evidence
7순위	james-stein	james-stein	one-hot
8순위	binary	binary	binary
9순위	ordinal	one-hot	frequency
10순위	frequency	leave one out	ordinal

분석 결과, 모델별로 베이지안 기반의 인코더 기법이 유용함. 하지만 높은 카디널리티에 유용하다는 것을 고려한다면, Logistic Regresssion에서는 One-Hot Encoding, Decision Tree에서는 Ordinal Encoding이 유용하다는 것을 알 수 있음. 또한, MLP에서는 Leave One Out Encoding이 유용하다는 것을 알 수 있음.

항목1
분석

04. 모델별 인코딩 분석

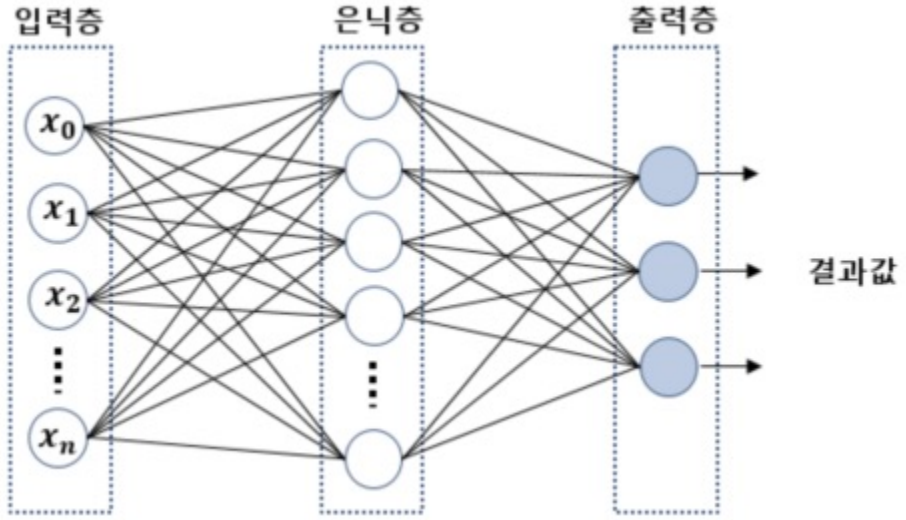
01 Logistic Regression에서 one-hot 인코딩 기법이 유용한 이유는?

one-hot 인코딩은 각 피처를 0 or 1로 표현하기 때문에 Logistic 같은 이분법에 적합함.

02 Decision Tree에서 Ordinal과 Frequency 인코딩 기법이 유용한 이유는?

Tree 모델은 순서를 고려하지 않으므로 Ordinal Encoding이 적합함.
Ordinal Encoding은 피처 개수를 증가시키지 않기 때문에 피처 중요도를 분할시키지 않아서 유용함.

03 MLP에서 Leave-One-Out 인코딩 기법이 유용한 이유는?



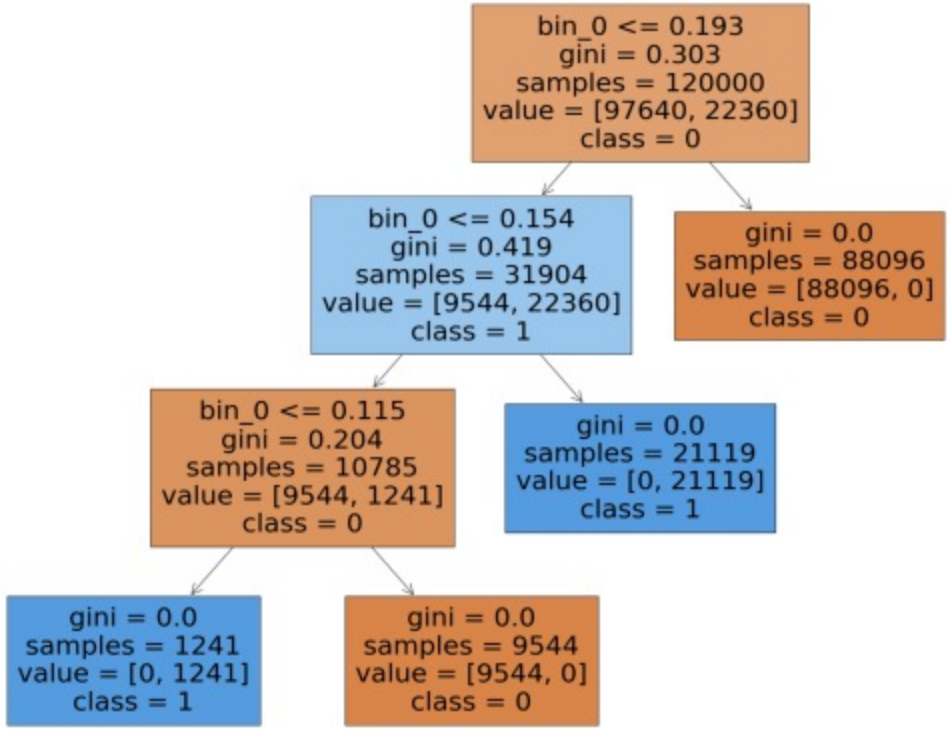
MLP모델의 '은닉층'은 우리가 알 수 없는 영역이고,
이상치가 존재할 수도 있음.

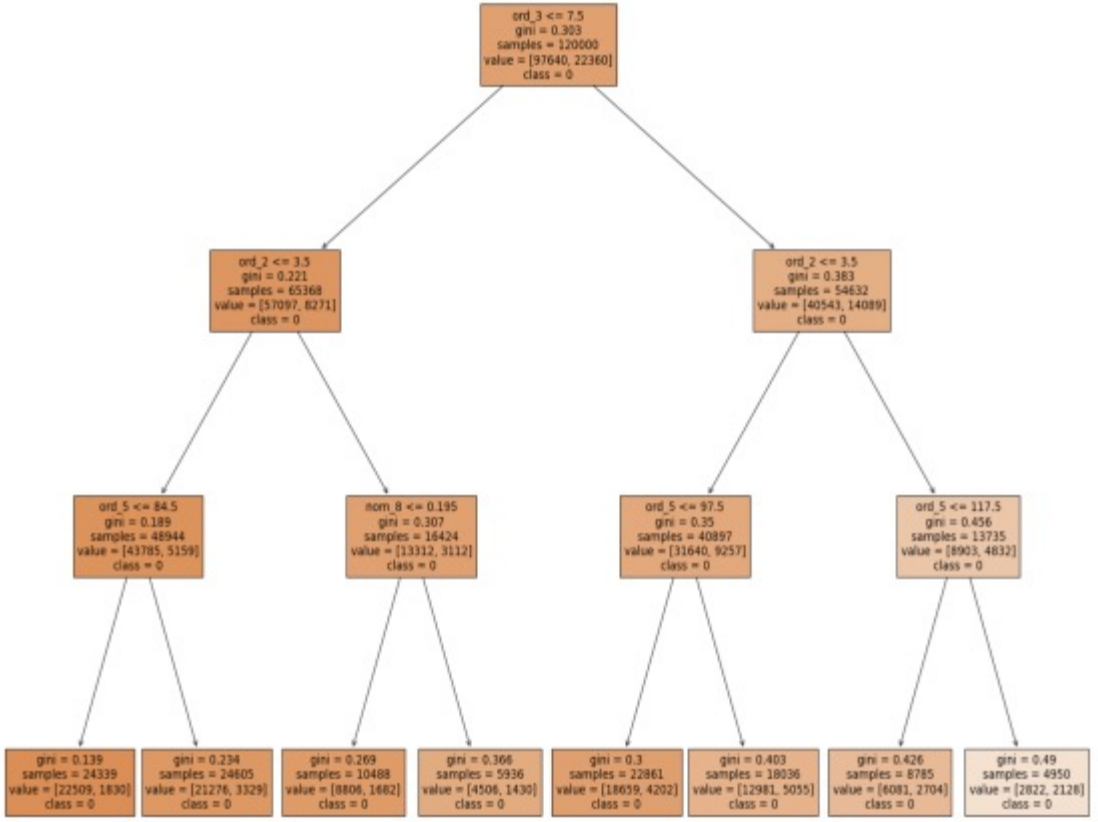
Leave-One-Out 인코딩 기법은 이상치의 영향력을
약화시키기 때문에 이 기법이 유용함.

항목1
분석

04. 모델별 인코딩 분석

04 Decision Tree에서 Leave One Out 인코딩 기법이 성능이 낮게 나온 이유는?





```
X_test_imp_loo['bin_0'].value_counts()
```

```
0.193371    363678
0.115067     36322
Name: bin_0, dtype: int64
```

왼쪽 Tree를 시각화한 결과, target 값이 모두 0으로 편향됨.

인코딩이 필요하지 않은 피처에 대해서도 Leave One Out 인코딩 기법을 적용하니 값이 편향된 것으로 추측.

오른쪽 Tree를 시각화한 결과, 인코딩이 필요하지 않은 피처에 대해서는 인코딩을 하지 않아 예측값이 편향되지 않고, 성능이 잘 나옴.

항목2
전략

05. 고유값 기준으로 Feature 분류

feature	dtype	count	feature	dtype	count
bin_0	int32	2	nom_6	object	1511
bin_1	int32	2	nom_7	object	222
bin_2	int32	2	nom_8	object	222
bin_3	object	2	nom_9	object	2202
bin_4	object	2	ord_0	int32	3
nom_0	object	3	ord_1	object	5
nom_1	object	6	ord_2	object	6
nom_2	object	6	ord_3	object	15
nom_3	object	6	ord_4	object	26
nom_4	object	4	ord_5	object	190
nom_5	object	1216	day	int32	7
			month	int32	12

(고유값 종류는 밑 페이지 참고)

binary features	Encoding
bin_0 ~ bin_2	0,1로 이루어져 있음
bin_3 ~ bin_4	T, Y을 1, F, N을 0으로 변경
nominal features	**Encoding**
nom_0 ~ nom_4	One-hot Encodig
nom_5 ~ nom_9	베이지안 기반 인코딩 시도
ordinal features	**Encoding**
ord_0	1,2,3으로 이루어져 있음
ord_1 ~ ord_2	숫자를 지정해서 인코딩
ord_3 ~ ord_5	Ordinal Encoding
cyclical features	**Encoding**
day,month	One-hot Encodig

feature의 데이터 타입과 고유값의 종류, 개수를 보고 4가지 종류의 feature로 나눈 다음, feature의 종류에 맞는 인코딩을 각각 진행

항목2 전략

06. Feature 종류별로 인코딩

feature	고유값	인코딩 방법
bin_0 ~ bin_2	0, 1	이미 숫자이므로 인코딩 X
bin_3	T, F	T를 1로, F를 0으로 직접 지정하여 인코딩
bin_4	Y, N	Y를 1로, N를 0으로 직접 지정하여 인코딩
nom_0 ~ nom_4	순서가 없고 개수가 적음	카디널리티가 낮으므로 One-hot Encoding
nom_5 ~ nom_9	순서가 없고 개수가 많음	카디널리티가 높으므로 베이지안 기반 인코딩 시도
ord_0	1, 2, 3	이미 숫자이므로 인코딩 X
ord_1	'Contributor', 'Master', 'Novice', 'Expert', 'Grandmaster'	'Novice': 0, 'Contributor': 1, 'Expert': 2, 'Master': 3, 'Grandmaster': 4로 직접 지정하여 인코딩
ord_2	'Cold', 'Warm', 'Freezing', 'Hot', 'Lava Hot', 'Boiling Hot'	'Freezing': 0, 'Cold': 1, 'Warm': 2, 'Hot': 3, 'Boiling Hot': 4, 'Lava Hot': 5 로 직접 지정하여 인코딩
ord_3 ~ ord_5	알파벳	순서가 있으므로 Ordinal Encoding
day	1, 2, 3, 4, 5, 6, 7	순서가 없고 카디널리티가 낮으므로 One-hot Encoding
month	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	순서가 없고 카디널리티가 낮으므로 One-hot Encoding

항목2
정리

07. nom_5 ~ nom_9 인코딩

01

실험 1의 결과, 베이지안 기반의 인코딩 기법이 카디널리티가 높은 범주형 변수 인코딩에 유용하다는 사실을 알게 되었음

02

nom_5 ~ nom_9 의 고유값의 개수가 각각 1216, 1511, 222, 222, 2202개로 카디널리티가 매우 높은 변수임을 알 수 있음

03

베이지안 기반의 인코딩 기법인 Target, James-Stein, Weight of Evidence, Leave One Out, M-Estimator, CatBoost 인코딩으로 nom_5 ~ nom_9를 인코딩함

04

인코딩한 결과를 Logistic Regression, Decision Tree, MLP 모델을 사용하여 모델링해서 성능을 비교함

항목2 분석

08. 최적의 조합

01 각 모델별(lg, dt, mpl순)로 **최고성능** 조합은
Weight of Evidence encoding, CatBoost encoding, Leave one out encoding이고,

02 각 모델별(lg, dt, mpl순)로 **최저성능** 조합은
Leave one out encoding, James-Stein encoding, Weight of Evidence encoding임

Encoder	Logistic	Decision Tree	MLP
Target	0.7432	0.64209	0.74425
Weight of Evidence	0.75655	0.64207	0.73943
Leave One Out	0.7421	0.66009	0.75151
James-Stein	0.74839	0.64163	0.74335
M-Estimator	0.74751	0.64211	0.74795
CatBoost	0.74225	0.66072	0.74509

성능 순위	logistic	decision tree	mlp
1순위	Weight of Evidence	catboost	leave one out
2순위	james-stein	leave one out	m-estimator
3순위	m-estimator	m-estimator	catboost
4순위	target	target	target
5순위	catboost	Weight of Evidence	james-stein
6순위	leave one out	james-stein	Weight of Evidence

M-Estimator은 Decision Tree 모델과 Logistic Regression 모델에서 세 번째로 높은 성능을 냈고,
Multu-Layer Perceoption 모델에서는 두 번째로 높은 성능을 기록함.
이에 따라, 중복도가 낮은 피쳐에는 M-Estimator을 적용하는 것이 좋다는 결론을 낼 수 있음.

09.결론

1. 모델별로 유용한 Categorical Feature Encoding 기법은?

모든 모델에서 베이지안 기반의 인코더 기법이 유용했고, 베이지안 인코더 기법을 제외하면 Logistic Regresssion에서는 One-Hot Encoding, Decision Tree에서는 Ordinal Encoding, MLP에서는 Leave One Out Encoding가 가장 유용했음

2. Categorical Feature 유형별로 유용한 Categorical Feature Encoding 기법은?

Binary feature는 직접 숫자를 지정해서 인코딩함. Nominal feature의 경우, 카디널리티가 낮을 때는 One-hot Encoding, 카디널리티가 높을 때는 베이지안 기반의 인코더 기법 중 M-Estimator Encoding이 가장 유용했음. Ordinal features의 경우에는 Ordinal Encoding이, Cyclical feature의 경우에는 One-hot Encoding이 가장 유용했음