

ENCODER (stringlarni raqamlarga o'tkazish)

- One - Hot Encoding — > One - Hot Encoding va Get_dummies
- Label Encoding
- Target Encoding
- Frequence Encoding

Encoder turlari va ishlatilishi

1. Label Encoding (Ordinal Encoding - tartibli)(Alphabet)

- **Qanday ishlaydi:** Har bir kategoriya **raqam** bilan almashtiriladi (0, 1, 2...).
- **Qachon ishlatiladi:** Kategoriyalar o'zaro tartibli bo'lsa (masalan, low, medium, high).
- **Misol:**
 - arduino
 - Copy code
 - Kategoriya: ['Low', 'Medium', 'High']
 - Kodlangan: [0, 1, 2]

Kamchilik:

- Modellar kategoriyalarni tartibli deb qabul qiladi, lekin ular tartibsiz bo'lsa, noto'g'ri natijalar bo'lishi mumkin.

2. One-Hot Encoding (Nominal -tartibsiz)

- **Qanday ishlaydi:** Har bir kategoriya uchun alohida ustun yaratiladi, va shu ustunda 1 yoki 0 yoziladi.
 - **Qachon ishlatiladi:** Tartibsiz kategoriyalar uchun.
 - **Misol:**
 - css
 - Copy code
 - ['Apple', 'Banana', 'Cherry'] →
 - Apple Banana Cherry
- | | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Kamchilik:

- Juda ko'p kategoriya bo'lsa, ustunlar soni oshib ketadi (sparsity muammosi).

3. Target Encoding (maqsadli nishon)(ikkinchi nomi MEAN encoding)

OUTPUT ga bevosita bog'liq bo'lgan encoding.

- **Qanday ishlaydi:** Har bir kategoriya uchun maqsadli nishon (target) qiymatlarning o'rtacha yoki boshqa statistikasi hisoblanadi.
- **Qachon ishlatiladi:**
 - Yuqori o'lchovli kategoriyali ma'lumotlar uchun.

- Misol:

```
Kategoriya: ['A', 'B', 'A', 'B', 'C']  
Target:      [100, 200, 150, 250, 300]
```

1. Target Encoding

Formula:

$$\text{Target Encoding}(k) = \frac{\text{Kategoriya } k \text{ uchun target yig'indisi}}{\text{Kategoriya } k \text{ uchun namunalar soni}}$$

Hisob-kitob:

- A:

$$\text{Target Encoding}(A) = \frac{100 + 150}{2} = \frac{250}{2} = 125$$

- B:


$$\text{Target Encoding}(B) = \frac{200 + 250}{2} = \frac{450}{2} = 225$$

- C:

$$\text{Target Encoding}(C) = \frac{300}{1} = 300$$

Natija:

```
yaml
```

 Copy code

```
A: 125, B: 225, C: 300
```

- **Kamchilik:**
 - Overfitting xavfi bor, chunki target ma'lumotlardan foydalanyapti.

4. Frequency Encoding (Nominal -taribsiz)

Objectlar nechi marta ishlayotganini tekshirib o'shani o'rtachasi olinadi.

- **Qanday ishlaydi:** Har bir kategoriya ma'lumot to'plamida necha marta uchraganiga qarab kodlanadi.
- **Qachon ishlatiladi:**
 - Juda katta kategoriyali ma'lumotlar uchun samarali.
- **Misol:**

2. Frequency Encoding

Formula:

$$\text{Frequency Encoding}(k) = \frac{\text{Kategoriya } k \text{ uchun namunalar soni}}{\text{Jami namunalar soni}}$$

Hisob-kitob:

- A:

$$\text{Frequency Encoding}(A) = \frac{2}{5} = 0.4$$

- B:


$$\text{Frequency Encoding}(B) = \frac{2}{5} = 0.4$$

- C:

$$\text{Frequency Encoding}(C) = \frac{1}{5} = 0.2$$

Natija:

yaml

 Copy code

A: 0.4, B: 0.4, C: 0.2

- **Kamchilik:**
- Ushbu kodlash kategoriyalar orasidagi o'zaro bog'liqlikni to'liq ifodalamaydi.

Qanday encoderni tanlash kerak?

1. **Tartibli kategoriya:** Label Encoding yoki Target Encoding.
2. **Tartibsiz kategoriya:** One-Hot Encoding.
3. **Katta kategoriyalar soni:** Frequency yoki Target Encoding.

Xulosa

- **Label** va **One-Hot Encoding** — asosiy klassik usullar.
- **Target** va **Frequency Encoding** — murakkab va katta kategoriyalar uchun mos.

One Hot Encoder o'zi 2 methodga bo'linadi. (OneHotEncoder va get_dummies)

OneHotEncoder vs pd.get_dummies

Xususiyat	OneHotEncoder (sklearn)	pd.get_dummies (pandas)
Kutubxona	scikit-learn	pandas
Interfeys	Machine Learning pipeline'lariga mos	DataFrame bilan ishlash uchun qulay
Natija turi	NumPy array yoki sparse matrix	Pandas DataFrame
NaN qiymatlar	Qo'llab-quvvatlanmaydi	NaN qiymatlarni avtomatik boshqaradi
Model bilan ishlash	Sklearn modellariga mos	To'g'ridan-to'g'ri modelda ishlatilmaydi
Bir nechta ustunlar	Qo'llab-quvvatlanadi	Qo'llab-quvvatlanadi
Kategoriyalarni boshqarish	O'qitish va sinov ma'lumotlari bilan bir xil kategoriya hosil qiladi	Kategoriyalarni avtomatik aniqlaydi
Qo'llanish joyi	ML pipeline va modellar bilan birga ishlatish uchun	Oddiy tahlillar va visualization uchun qulay

Qachon qaysi biridan foydalanish kerak?

1. **OneHotEncoder:**
 - Machine Learning pipeline'larida, scikit-learn modellarida ishlatiladi.
 - NumPy array yoki sparse matrixni yaratadi, bu ML modellar bilan samarali ishlaydi.
2. **pd.get_dummies:**
 - Pandas DataFrame bilan ishlayotganingizda.
 - Oddiy tahlil yoki vizualizatsiya uchun ma'lumotlarni One-Hot Encoding qilishda.

Ordinal va Nominal kategoriyalar uchun encodinglar jadvali

Kategoriya turi	Tavsifi	Mos encodinglar	Nega ishlatiladi?	Misol
Ordinal Category	Tartibli: Kategoriyalar orasida tartib mavjud.	- Label Encoding-Ordinal Encoding	Tartibni saqlab qolish uchun.	['Low', 'Medium', 'High'] → [0, 1, 2]
Nominal Category	Tartibsiz: Kategoriyalar orasida tartib yo'q.	- One-Hot Encoding-Frequency Encoding-Target Encoding	Bog'liqlikni oldini olish yoki ko'p kategoriyalarni optimallashtirish uchun.	One-Hot: ['Red', 'Green', 'Blue'] → [[1,0,0],[0,1,0],[0,0,1]] Frequency: ['Red', 'Green', 'Blue', 'Red'] → {'Red': 2, 'Green': 1, 'Blue': 1} Target: ['Red', 'Green', 'Blue'] → {'Red': 100, 'Green': 200, 'Blue': 150}

One Hot Encoder bilan Label Encoder tanlash

Qaysi Encoder ni tanlashdan oldin Cardinality ni aniqlab olamiz. Cardinality bu bitta feature (xususiyat=ustun) ni ichidagi classlar soni.

Masalan rang degan feature bor va uning ichida 5 xil rang (class)lar bor. Cardinalitysi 5 ga teng.

Agar cardinality = > 5 dan bo'lsa Label Encoding Agar cardinality = < 5 bo'lsa One Hot Encodingdan foydalaniladi.