

7-dars (Report)

Data Preprocessing:

• Handling missing values

- mean
- mode
- median
- fixed
- drop

Basic

• Encoding (method)

- One-hot
- Label Encoding

- Frequency
- Target
- Ordinal

• Scaling

* Frequency encoding

- Taroqlanishlar sonini umumiy elementlar soniga nisbati bilan almashtiriladi.

Column	Animal	Animal_freq
Num	cat	(cat) 0.5
Num	cat	(cat) 0.5
Num	dog	(dog) 0.25
Num	crocodile	(crocodile) 0.25

(Nima uchun?)

chunki qatorlar soni 4 ta va 'cat' 2 ta, crocodile 1 ta, dog 1 ta taligi uchun
 $\frac{2}{4} \rightarrow$ qatorlar soni
 $\frac{1}{4}$ dog; crocodile $\rightarrow \frac{1}{4}$

* Target encoding (output)

Column1	Animal	Column2
1.5	cat	0
3.6	cat	1
4.2	dog	1
7.1	crocodile	1

Column1

Animal

(Target)
Column2

1.5

0.5

0

3.6

0.5

1

4.2

1

1

7.1

1

1

Target encodingda harfli Columnlarda quyidagilarni hisoblaydi:

Animal	Column2
cat	0
cat	1
dog	1
crocodile	1

2 ta catning ro'parasidagini qo'shsak ($0+1=1$) \Rightarrow uni (2 ta) catga bilsak ($\frac{1}{2} \Rightarrow 0.5$)
 dogning ro'parasida (1) va dog (1) taligi uchun $\Rightarrow (1)$
 crocodile ro'parasida (1) va crocodile (1) taligi uchun $\Rightarrow (1)$

* Ordinal encoding

- Ketma-ketlikka ushbu sonlar bilan almashtiriladi: quyidagilardan boshlab katta tomonga.

Column1	size	Column2
50	large	1.0
35	small	3.0
75	extra large	0.0
42	medium	2.0
54	large	1.0
71	extra large	0.0

\Rightarrow

Size
S
XS
L
M

Ordinal encoding

Encoding
2
1
4
3

Size kichigidan katta tomonga:
 $XS \rightarrow S \rightarrow M \rightarrow L \rightarrow XL(S)$

LABEL \Rightarrow alfavit bo'yicha ro'yxatlashiladi
 ketma-ketligi qog'

Ordinal \Rightarrow ketma-ketlikni uzmaslik kerak.
 kattadan \rightarrow kichikdan / yoki / kichikdan \rightarrow kattaga

\Rightarrow ita datasetda faqat birinchi qatnash kerak.

For loop

① Data types

② Operations

③ Conditions (if/else)

④ For loop (Data ni avtomatlashtirish uchun ishlatiladigan kodlar bloki, jarayonni avtomatlashtirish xislati)

```
a = [1, 2, 3]
```

```
print(a[0])
```

```
print(a[1])
```

```
print(a[2])
```

V 0.0s

1
2
3

⇒ for i in [1, 2, 3]
 print(i)

for i ← keywords
0.0s
1
2
3

1) a = [1, 2, 3]

```
for i in a:  
    print(a)
```

1
2
3

```
for element in a:  
    print(element)
```

1
2
3

```
b = ['Hello', 'Hi', 'Bye']
```

```
for i in b:  
    print(len(i))
```

5
2
3

len() → nechta harf dan iboratligini ko'rish

```
for i in a:  
    print(a+i)
```

2
3
4

(oralig)
for i in range(1, 10) ⇒ 1 dan 9 gacha sonni oladi.
for i in range(1, 11) ⇒ 1 dan 10 gacha sonni oladi.

1) 1 dan 10 gacha faqat juft sonlarni kiritish uchun:

```
for i in range(1, 11):  
    if i % 2 == 0:  
        print(i)
```

2
4
6
8

2) 1 dan 10 gacha to'g'ri sonlar uchun:

```
for i in range(1, 11):  
    if i % 2 != 0:  
        print(i)
```

1
3
5
7
9

3) for i in range(1, 11, 2):
 print(i)

1
3
5
7
9

⇒ Range nima? oralig' degani
range(start, stop, step)

2 ta dan g'adan tashqari
deganmiz
oldingi misolda:

DATA PREPROCESSING + FOR LOOP

1) Missing + for loop

df.columns # ustundagi barcha ustunlar nomi
print(df.dtypes)

★ for col in df.columns:

if df[col].isnull().any():

if df[col].dtype == 'string':

df[col] = df[col].fillna(df[col].mode()[0], inplace=True)

else:

df[col] = df[col].fillna(df[col].mean(), inplace=True)

⇒ Agar "nan" dan oldin teng (=) ishorasi
bolsa "mean" paytida "inplace=True" qo'shasan
bolsa di.

2) Encoding + for loop

⇒ from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

encoder

1) df['Ustun nomi'] = encoder.fit_transform(df['Ustun nomi'])

2) df.unique

dummies = pd.get_dummies(df['Ustun nomi'], prefix='Ustun nomi', dtype=int)

dummies (Shift + Enter)

df = pd.concat([df.drop(columns=['Ustun nomi']), dummies], axis=1)

★ for loopni encodingda tadbiq qilish

for col in df.columns:

if df[col].dtype == 'string': # string yoki object bitadi

if df[col].nunique() < 5:

dummies = pd.get_dummies(df[col], prefix=col, dtype=int)

df = pd.concat([df.drop(columns=col), dummies], axis=1)

else:

df[col] = encoder.fit_transform(df[col])