

4 - Dars report

Encoding slayomi \Rightarrow

Frequency encoding

Takrorlanishlar sonini umumiy elementlar soniga nisbati b/h almashtiriladi.

Target encoding

Mos target qiymatning o'rtacha arifmetikiga kora encoding qiladi.

Ordinal encoding

Ketma ketligiga kora sonlar b/h almashtiriladi.

for loop

- 1) Data types
- 2) Operations
- 3) Conditions
- 4) for loop

for loop → Yarayonni avtomatlashtirish uchun ishlatiladigan kodlar bloki. for loop biqa qulaylik yaratib beradi. Misol uchun:

$a = [1, 2, 3]$

`print(a[0])`

`print(a[1])`

`print(a[2])`

⇒ Bu for loopsiz qilinsa

`for i in [1, 2, 3]:`
`print(i)`

natija = 1, 2, 3 chiqadi.

for bu yerda "key words",
"i" esa har bir sondi ichi-
ni oqib chiqib natijani chi-
qaradi. Yana code b/k ko'rib
chiqamiz ⇒

⇒ $a = 1, 2, 3$

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

1

2

3

`for i in a:`
`print(i+1)`

2

3

4

for i in a:
print(i**2)

2
4
6

B = ['Hello', 'HI', 'bye']

for i in B:
print(i)

Hello
HI
bye

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

1 bu cod wa'lar bir tonda
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

if = condition oralastik
has i'lastak kodak:
list uchun just contain
sora sh: Code =>

for i in range(1, 10):
if i % 2 == 0:
print(i)

toylarini topish codi:

for i in range(1, 10):
if i % 2 != 0:
print(i)

1
3
5
7
9


```
for i in range(1, 10, 2):
    print(i)
```

1
3
5
7
9

"range" ichiligi cools
range(start, stop, step)

start → 1 da davom et

stop → 10 da toxta

step → har 2 keda sakra

Endi ML ga ta'dbiq qilamiz

1) datasetni yuklaymiz

```
import pandas as pd
```

```
df = pd.read_csv('cory')
```

df.info()

qilib ma'lumot olamiz.

Data preprocessing +
for loop

1) Missing + for loop

kishub qolgan qiymatlarini
aniqlaymiz, agar bo'lmasa
datasetdan o'chirib Ctrl+S

ni borib save qilamiz va

datasetni qayta yuklaymiz!

Endi oldingi usul bika to-
ldirib olamiz ⇒ objda Miss!

```
#df['us'].fillna().mode()[0], inplace=T)
```

Endi kishub qolgan qiymatni

cool: b/c tekstiv's olavix.
 ustun olag: qinuat wol bo-
 logan bolad: fud: Moan =>
 ni: haw qitib olavix va
 wol qitib olavix, keyin
 for loop b/c qibad.
 iyo qilavix chunk: type
 haw q' kerol. End: outmost
 lastivis oon yot b/c
 hanna qinuatoni: bolotia-
 nuix. cool:
 Hanne ustun haw: bolotia
 uchun bix bu cool's har-
 y olar foydalanuvix:
 # of columns
 keyin

for col in of.columns:
 if of[col].isnull().any():
 if of[col].dtype == 'object':
 of[col].fillna(.mode()[0], in=T)
 else:
 of[col].fillna(.mean(), in=T)
 End: bu cool nuix hanna
 fudub qolotivix qinuat-
 ni: bolotiad: fag bolaga
 holatlar bolta # drop
 bayruqin: haw berak bol-
 tad.
 # Encoding + for loop
 bix olavix haw int float

ya potdizivaiet. Endi yuki.
 lab otvori coot.
 from stream.py ... input =>
 => LabelEncoder

encoder = LabelEncoder

encoder

bit computer kulukiti

uchen numerica ginyetga
otkazani

of[us] = encoder.fit_transform()

info gisak object => int
 ga otgar bolat. Endi
 One-hot coding gilausi

df.unique()
 gilib eng mayda gilyatli.
 utrud. obestliti otib
 int / float gilausi: cod

dummies = pd.get_dummies(df['1'], prefix='cat',
 => dtype=int)

dummies

Endi original datasetga qoshil
vix

df = pd.concat([df, dummies], axis=1)
 => dummies otisil

info gisak qoshilgan bolat.

ЧИСЛО

МЕСЯЦ

ГОД

ПН

БТ

СР

ЧТ

ПТ

СБ

ВС

8 each: for loop for previous:

9 for col in df.columns:

10 if df[col].dtype == 'object':

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

12 dummies = pd.get_dummies(df[col], prefix =

13 df = pd.concat([df, dummies], axis=1)

14 else:

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

16 forget obj: when it's

17 ok: again obj: because

18 with get_dummies we have

19 the same number of rows

20 and just change

ЧИСЛО

МЕСЯЦ

ГОД

ПН

БТ

СР

ЧТ

ПТ

СБ

ВС

8 get_dummies when it's ok:

9 you can

10 post = 0

11 for i in range(1, 10):

12 if i % 2 == 0:

13 post = post + i

14 print(i)

15 print(post)

16 2

17 4

18 6

19 8

20 10

21 12

22 14

23 16

24 18

25 20

26 22

27 24

28 26

29 28

30 30

31 32