

* Encoding Categorical Variable

↳ Feature Eng → Feature transform

feature Scaling Encoding Categorical.

(data) → Numerical

↳ Categorical Data

Ordinal

Relation ✓

→ Marks
→ Review

Excellent ↑
good ↑
?

Nominal Categorical

↓
there isn't any
order
or relation

State

College Branch

↳

Encoding Categorical

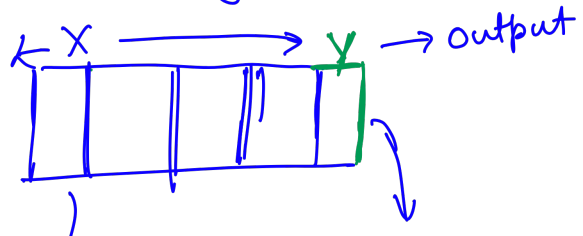
→ Ordinal Encoding

→ One hot Encoding (Nominal Encoding)

Label Encoding

Label Encoding

→ Input



Categorical
Value

if x we have
any

if "y"
we have
Categorical
Value

Mostly in Classification

how we use
Label Encoding

Data

Education

H S
UG
PG
UG

we need to transform

check for is it

- Nominal
- Ordinal

there is a Relation on

so it will be
Ordinal.

$H S < U G < P G$

$\left\{ \begin{array}{l} P G = 2 \\ U G = 1 \\ H S = 0 \end{array} \right\}$

```
[1] import numpy as np
import pandas as pd
✓ 2.7s

[2] df = pd.read_csv('customer.csv')
✓ 0.0s

[3] df.head(5)
✓ 0.0s
...
age  gender  review  education  purchased
0    30   Female  Average    School        No
1    68   Female   Poor     UG          No
2    70   Female   Good     PG          No
3    72   Female   Good     PG          No
4    16   Female  Average     UG          No

[4] df = df.iloc[:,2:]
✓ 0.0s

[6] df.head(2)
✓ 0.0s
...
review  education  purchased
0  Average    School        No
1    Poor      UG          No
```

Target Column



It Says that when we recommended
Any Products

Did they bought → yes
 → NO



Step 1 → Need to Identify
 → What type of Categorical
 Column
 It is

On hot Encoder

gender → Nominal Categorical
Column

Male → female → No Relation

Ordinal Encoder

Review → Orderly
Good, Average, Poor →

Education → Orderly.

Label Encoder

Purchase → yes / NO → Not depended
Nominal

	age	gender	review	education	purchased
0	30	Female	Average	School	No
1	68	Female	Poor	UG	No
2	70	Female	Good	PG	No
3	72	Female	Good	PG	No
4	16	Female	Average	UG	No

Process → It is Complex



If we need to apply On hot Coding
to gender

Same for review
Education

We need to remove gender
And then Apply to

And then Purchase

On hot Encoding

then Attach



To Complex

→ So we have one
Class
in Sciklearn



In this
Alag Alag

Pipeline ke

through Column transform

ke through Ek Sath karudegi

Column transform

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(df.iloc[:,0:2],df.iloc[:,1],test_size=0.2)
```

✓ 0.0s

```
x_train.head(2)
```

✓ 0.0s

	review	education
0	Average	School
29	Average	UG

```
from sklearn.preprocessing import OrdinalEncoder
```

```
oe =OrdinalEncoder(categories=[['Poor','Average','Good'], ['School','UG','PG']])
```

✓ 0.0s

```
oe.fit(x_train)
```

✓ 0.0s

```
OrdinalEncoder
OrdinalEncoder(categories=[['Poor', 'Average', 'Good'], ['School', 'UG', 'PG']])
```

```
x_train=oe.transform(x_train)
```

```
x_test=oe.transform(x_test)
```

✓ 0.0s

} → Test/Train Split

giving high Value

We fit in train

transform → Don't pe

Train ← Karu
Test hai

We need to make Sure → for Output Column Only

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
le.fit(y_train)
```

✓ 0.0s

```
LabelEncoder
LabelEncoder()
```

```
le.classes_
```

✓ 0.0s

```
array(['No', 'Yes'], dtype=object)
```

```
y_train= le.transform(y_train)
```

```
y_test= le.transform(y_test)
```

✓ 0.0s

```
y_train
```

✓ 0.0s

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