Ordinal Encoding

What is Ordinal Encoding:

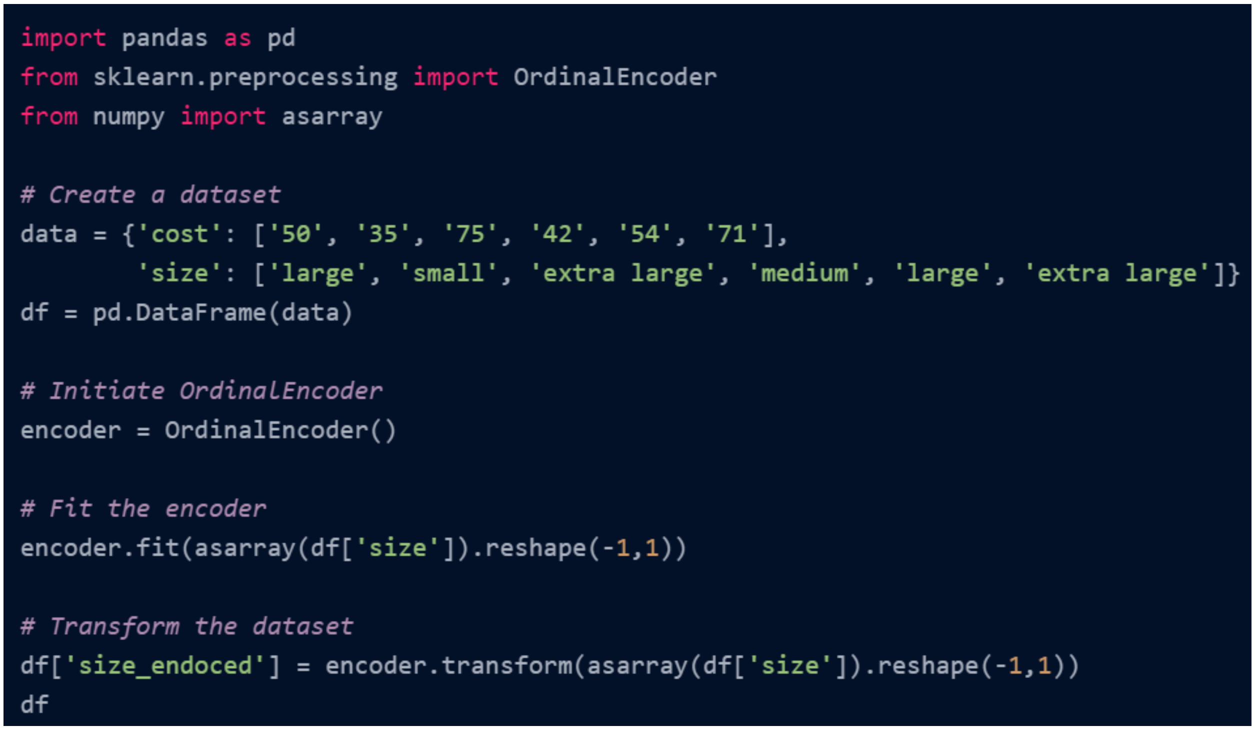
Ordinal encoding is a technique that is used to transform categorical variables into a numerical format by assigning a unique value to each of its categories. It is also referred to as Label Encoding. For example, we have customer feedback data based on a survey or online feedback mechanism. It contains categories - very dissatisfied, dissatisfied, neutral, satisfied, and very satisfied. To encode this variable using ordinal encoding, we can assign numerical values as mentioned below –

- very dissatisfied – 1
- dissatisfied – 2
- neutral – 3
- satisfied – 4
- very satisfied – 5

Ordinal encoding assumes that categories in categorical variables have clear, natural, and intrinsic ordering to their categories. It does not work for nominal categorical variables as no relationship exists between categories of a nominal variable. In our previous example, we encoded the categorical variable by assigning the lowest numerical value of 1 to the very dissatisfied category and the highest value of 5 to the very satisfied category. This way, we were able to preserve the natural ordering of the categories - very dissatisfied < dissatisfied < neutral < satisfied < very satisfied was retained in 1 < 2 < 3 < 4 < 5. Suppose we have another categorical variable, which contains red, blue, and green categories. We can encode this variable using ordinal encoding by assigning 1 to red, 2 to blue, and 3 to green, but it may lead to incorrect results. As encoded values have a natural ordering between them - 1 < 2 < 3 will be there, but red < blue < green does not exist.

Example: Encoding Categorical Data using Ordinal Encoding

Let’s understand how you can apply ordinal encoding to categorical features using Python libraries. We will use the OrdinalEncoder class provided by the sklearn library.



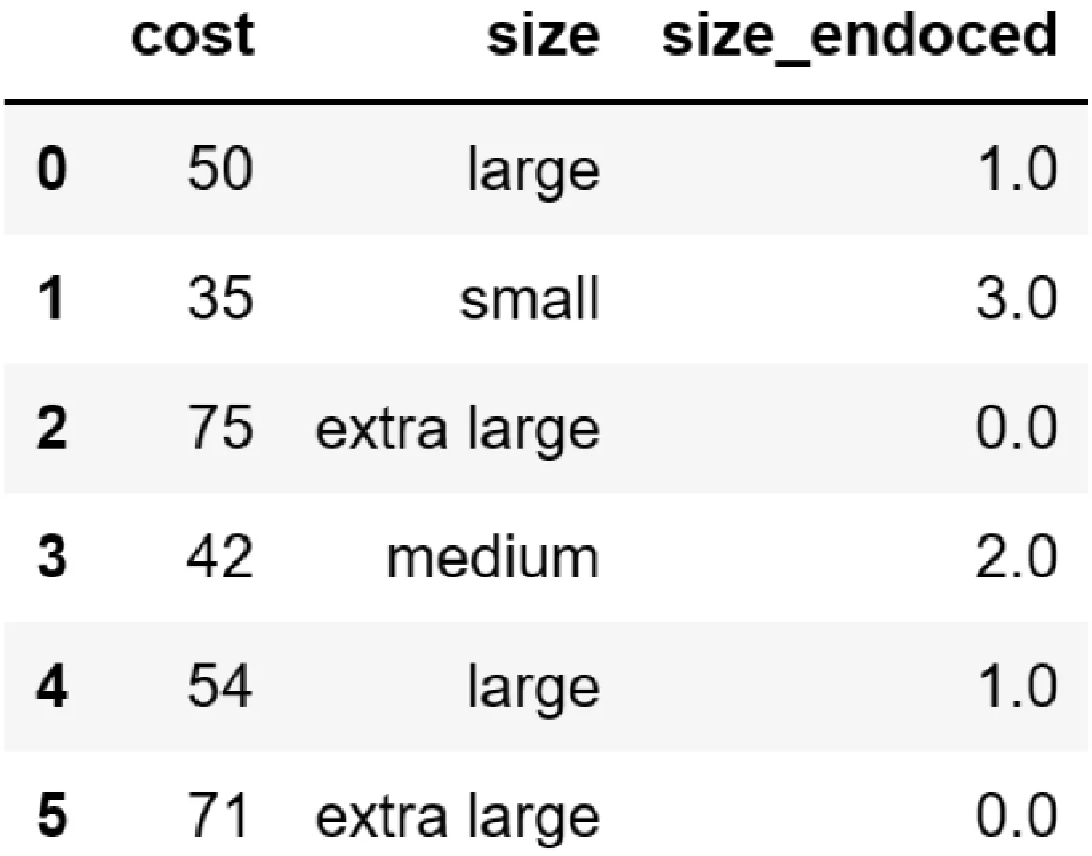
```
import pandas as pd
from sklearn.preprocessing import OrdinalEncoder
from numpy import asarray

# Create a dataset
data = {'cost': ['50', '35', '75', '42', '54', '71'],
        'size': ['large', 'small', 'extra large', 'medium', 'large', 'extra large']}
df = pd.DataFrame(data)

# Initiate OrdinalEncoder
encoder = OrdinalEncoder()

# Fit the encoder
encoder.fit(asarray(df['size']).reshape(-1,1))

# Transform the dataset
df['size_endoced'] = encoder.transform(asarray(df['size']).reshape(-1,1))
df
```



	cost	size	size_endoced
0	50	large	1.0
1	35	small	3.0
2	75	extra large	0.0
3	42	medium	2.0
4	54	large	1.0
5	71	extra large	0.0