Encoding

Converting Discrete Categorical Variable to Discrete Numerical Variable

In [1]:

```python
import numpy as np
import pandas as pd
```

> **Ordinal Data** (Ex : Shirt size)
>
> - If categories are ordinal, then apply ordinal encoding on that feature

In [2]:

```python
df1 = pd.DataFrame({"size": ["small","medium","high"]})
df1
```

Out[2]:

	size
0	small
1	medium
2	high

In [3]:

```python
df1["size"].value_counts()
```

Out[3]:

```
small     1
medium    1
high      1
Name: size, dtype: int64
```

Label Encoding

- In Label encoding, each category is assigned a value from 1 to N, where N is the number of categories of that feature.
- It converts to numeric as per alphabetical order

In [4]:

```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df1["size_le_enc"] = le.fit_transform(df1["size"])
df1
```

Out[4]:

	size	size_le_enc
0	small	2
1	medium	1
2	high	0

Ordinal Encoding

- convert to numeric as per given order in the function (ascending order)

In [5]:

```python
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder(categories=[["small","medium","high"]])
df1["size_ord_enc"] = oe.fit_transform(df1[["size"]])
df1
```

Out[5]:

	size	size_le_enc	size_ord_enc
0	small	2	0.0
1	medium	1	1.0
2	high	0	2.0

Feature Mapping

- convert to numeric, by mapping each category to a value

In [6]:

```python
df1['size_fm_pan'] = df1['size'].map({'small': 0,'medium':1,'high': 2})
df1
```

Out[6]:

	size	size_le_enc	size_ord_enc	size_fm_pan
0	small	2	0.0	0
1	medium	1	1.0	1
2	high	0	2.0	2

> **Nominal Data** (Ex : City names)
>
> - If categories are nominal, then apply nominal encoding on that feature

In [7]:

```python
df = pd.DataFrame({"town": ["Chennai","Bangalore","Hyderabad"]})
df
```

Out[7]:

	town
0	Chennai
1	Bangalore
2	Hyderabad

In [8]:

```python
df["town"].value_counts()
```

Out[8]:

```
Chennai      1
Bangalore    1
Hyderabad    1
Name: town, dtype: int64
```

OneHotEncoding

In [9]:

```python
from sklearn.preprocessing import OneHotEncoder
enc = OneHotEncoder(drop='first')
enc_df = pd.DataFrame(enc.fit_transform(df[['town']]).toarray(),columns=["Chennai","Hyderabad"])
df_ohe = pd.concat([df,enc_df],axis='columns')
df_ohe
```

Out[9]:

	town	Chennai	Hyderabad
0	Chennai	1.0	0.0
1	Bangalore	0.0	0.0
2	Hyderabad	0.0	1.0

Dummy Encoding

In [10]:

```
dum = pd.get_dummies(df["town"],drop_first=True)
df_dum = pd.concat([df,dum],axis='columns')
df_dum
```

Out[10]:

	town	Chennai	Hyderabad
0	Chennai	1	0
1	Bangalore	0	0
2	Hyderabad	0	1