Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

```
df =
pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Custom
er%20Purchase.csv')
```

Get Information of Dataframe

```
df.head()
```

```
   Customer ID  Age  Gender Education   Review Purchased
0         1021   30  Female    School  Average        No
1         1022   68  Female        UG     Poor        No
2         1023   70  Female        PG     Good        No
3         1024   72  Female        PG     Good        No
4         1025   16  Female        UG  Average        No
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Customer ID  50 non-null     int64 
 1   Age          50 non-null     int64 
 2   Gender       50 non-null     object
 3   Education    50 non-null     object
 4   Review       50 non-null     object
 5   Purchased    50 non-null     object
dtypes: int64(2), object(4)
memory usage: 2.5+ KB
```

```
df.isnull().sum()
```

```
Customer ID    0
Age            0
Gender         0
Education      0
Review         0
Purchased      0
dtype: int64
```

```
df.columns
```

```
Index(['Customer ID', 'Age', 'Gender', 'Education', 'Review',
'Purchased'], dtype='object')
```

Setting X and y

```
X= df[['Age', 'Gender', 'Education', 'Review']]
```

```
y=df['Purchased']
```

```
X[:]
```

```
    Age  Gender Education   Review
0    30  Female    School  Average
1    68  Female        UG     Poor
2    70  Female        PG     Good
3    72  Female        PG     Good
4    16  Female        UG  Average
5    31  Female    School  Average
6    18    Male    School     Good
7    60  Female    School     Poor
8    65  Female        UG  Average
9    74    Male        UG     Good
10   98  Female        UG     Good
11   74    Male        UG     Good
12   51    Male    School     Poor
13   57  Female    School  Average
14   15    Male        PG     Poor
15   75    Male        UG     Poor
16   59    Male        UG     Poor
17   22  Female        UG     Poor
18   19    Male    School     Good
19   97    Male        PG     Poor
20   57  Female    School  Average
21   32    Male        PG  Average
22   18  Female        PG     Poor
23   96  Female    School     Good
24   16  Female        PG  Average
25   57  Female    School     Good
26   53  Female        PG     Poor
27   69  Female        PG     Poor
28   48    Male    School     Poor
29   83  Female        UG  Average
30   73    Male        UG  Average
31   22  Female    School     Poor
32   92    Male        UG  Average
33   89  Female        PG     Good
34   86    Male    School  Average
35   74    Male    School     Poor
36   34  Female        UG     Good
```

```
37   94    Male      PG  Average
38   45  Female  School     Good
39   76    Male      PG     Poor
40   39    Male  School     Good
41   23    Male      PG     Good
42   30  Female      PG     Good
43   27    Male      PG     Poor
44   77  Female      UG  Average
45   61    Male      PG     Poor
46   64  Female      PG     Poor
47   38  Female      PG     Good
48   39  Female      UG     Good
49   25  Female      UG     Good
```

Ordinal Encoder for rank categorical data

```
from sklearn.preprocessing import OrdinalEncoder
oe=OrdinalEncoder()
X[['Gender', 'Education', 'Review']]=oe.fit_transform(X[['Gender',
'Education', 'Review']]
#X[:,1:]=oe.fit_tranform(X[:,1:])
X
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:3678:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  self[col] = igetitem(value, i)
```

```
    Age  Gender  Education  Review
0    30     0.0        1.0     0.0
1    68     0.0        2.0     2.0
2    70     0.0        0.0     1.0
3    72     0.0        0.0     1.0
4    16     0.0        2.0     0.0
5    31     0.0        1.0     0.0
6    18     1.0        1.0     1.0
7    60     0.0        1.0     2.0
8    65     0.0        2.0     0.0
9    74     1.0        2.0     1.0
10   98     0.0        2.0     1.0
11   74     1.0        2.0     1.0
12   51     1.0        1.0     2.0
13   57     0.0        1.0     0.0
14   15     1.0        0.0     2.0
15   75     1.0        2.0     2.0
```

```
16   59     1.0         2.0     2.0
17   22     0.0         2.0     2.0
18   19     1.0         1.0     1.0
19   97     1.0         0.0     2.0
20   57     0.0         1.0     0.0
21   32     1.0         0.0     0.0
22   18     0.0         0.0     2.0
23   96     0.0         1.0     1.0
24   16     0.0         0.0     0.0
25   57     0.0         1.0     1.0
26   53     0.0         0.0     2.0
27   69     0.0         0.0     2.0
28   48     1.0         1.0     2.0
29   83     0.0         2.0     0.0
30   73     1.0         2.0     0.0
31   22     0.0         1.0     2.0
32   92     1.0         2.0     0.0
33   89     0.0         0.0     1.0
34   86     1.0         1.0     0.0
35   74     1.0         1.0     2.0
36   34     0.0         2.0     1.0
37   94     1.0         0.0     0.0
38   45     0.0         1.0     1.0
39   76     1.0         0.0     2.0
40   39     1.0         1.0     1.0
41   23     1.0         0.0     1.0
42   30     0.0         0.0     1.0
43   27     1.0         0.0     2.0
44   77     0.0         2.0     0.0
45   61     1.0         0.0     2.0
46   64     0.0         0.0     2.0
47   38     0.0         0.0     1.0
48   39     0.0         2.0     1.0
49   25     0.0         2.0     1.0
```

Display Encoded Categories

```
oe.categories_
```

```
[array(['Female', 'Male'], dtype=object),
 array(['PG', 'School', 'UG'], dtype=object),
 array(['Average', 'Good', 'Poor'], dtype=object)]
```

Reverse Encode Integer array to Categorical Features

```
oe.inverse_transform([[0,0,0]])
```

```
array([[15, 'Female', 'PG', 'Average']], dtype=object)
```

```
oe.inverse_transform([[1,1,1]])
```

```
array([['Male', 'School', 'Good']], dtype=object)
```

```
oe.inverse_transform([[1,2,2]])
```

```
array([['Male', 'UG', 'Poor']], dtype=object)
```

Encoding Categorical features as an integer array of your choice

```
X=df[['Gender', 'Education', 'Review']]
oe=OrdinalEncoder(categories=[['Male','Female'],['School','UG','PG'],
['Poor','Average','Good']])
X=oe.fit_transform(X)
oe.categories_
```

```
[array(['Male', 'Female'], dtype=object),
 array(['School', 'UG', 'PG'], dtype=object),
 array(['Poor', 'Average', 'Good'], dtype=object)]
```