📦 Import Required Libraries

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

```
df=pd.read_csv('/content/customer.csv')
```

```
df.sample(5)
```

	age	gender	review	education	purchased
10	98	Female	Good	UG	Yes
33	89	Female	Good	PG	Yes
4	16	Female	Average	UG	No
27	69	Female	Poor	PG	No
31	22	Female	Poor	School	Yes

```
df=df.iloc[:,2:]
```

🔀 Train–Test Split

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(df.drop('purchased',axis=1),
                                               df['purchased'],
                                               test_size=0.2,
                                               random_state=0)
```

```
x_train.shape
```

```
(40, 2)
```

```
y_train.shape
```

```
(40,)
```

🔢 Categorical Encoding (Ordinal Encoding)

```
from sklearn.preprocessing import OrdinalEncoder
oe=OrdinalEncoder(categories=[['Poor','Average','Good'],['School','UG','PG']])
```

```
oe
```

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

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

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

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

```
x_train
```

```
array([[2., 2.],
       [0., 0.],
       [0., 2.],
       [1., 0.],
       [2., 0.],
       [0., 0.],
       [0., 2.],
       [0., 2.],
       [2., 1.],
       [1., 1.],
       [0., 1.],
       [1., 1.],
       [1., 1.],
       [0., 1.],
       [2., 2.],
       [1., 0.],
       [0., 2.],
       [1., 1.],
       [1., 0.],
       [2., 0.],
       [1., 0.],
       [0., 1.],
       [2., 0.],
       [2., 1.],
       [0., 1.],
       [0., 0.],
       [1., 2.],
       [1., 2.],
       [2., 0.],
       [2., 0.],
       [2., 1.],
       [1., 2.],
       [0., 2.],
       [2., 1.],
       [0., 2.],
       [0., 2.],
       [2., 2.],
       [1., 0.],
       [2., 2.],
       [1., 1.]])
```

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

```
x_test
```

```
array([[0., 0.],
       [2., 1.],
       [2., 1.],
       [2., 2.],
       [2., 2.],
       [0., 2.],
       [2., 0.],
       [0., 0.],
```

```
        [0., 2.],
        [1., 1.]])
```

🏷️ Categorical Encoding (Label Encoding)

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

```
le=LabelEncoder()
```

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

```
▾ LabelEncoder
LabelEncoder()
```

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

```
y_train
```

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

```
le.classes_
```

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

🏷️ Categorical Encoding (One-Hot Encoding)

```
car=pd.read_csv('/content/cars.csv')
```

```
car['brand'].value_counts()
```

brand	count
Maruti	2448
Hyundai	1415
Mahindra	772
Tata	734
Toyota	488
Honda	467
Ford	397
Chevrolet	230
Renault	228
Volkswagen	186
BMW	120
Skoda	105
Nissan	81
Jaguar	71
Volvo	67
Datsun	65
Mercedes-Benz	54
Fiat	47
Audi	40
Lexus	34
Jeep	31
Mitsubishi	14
Land	6
Force	6
Isuzu	5
Ambassador	4
Kia	4
MG	3
Daewoo	3
Ashok	1
Opel	1
Peugeot	1

dtype: int64

```
car.sample(5)
```

	brand	km_driven	fuel	owner	selling_price
7988	Maruti	32000	Diesel	First Owner	1025000
4308	Audi	55000	Petrol	Second Owner	730000
319	Maruti	47000	Petrol	First Owner	409999
4868	Toyota	60000	Petrol	Second Owner	150000
8114	Maruti	73000	Petrol	First Owner	200000

```
car.iloc[:,0:4]
```

	brand	km_driven	fuel	owner
0	Maruti	145500	Diesel	First Owner
1	Skoda	120000	Diesel	Second Owner
2	Honda	140000	Petrol	Third Owner
3	Hyundai	127000	Diesel	First Owner
4	Maruti	120000	Petrol	First Owner
...	...	...	...	...
8123	Hyundai	110000	Petrol	First Owner
8124	Hyundai	119000	Diesel	Fourth & Above Owner
8125	Maruti	120000	Diesel	First Owner
8126	Tata	25000	Diesel	First Owner
8127	Tata	25000	Diesel	First Owner

8128 rows × 4 columns

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(car.iloc[:,0:4],
                                               car.iloc[:,:-1],
                                               test_size=0.2,
                                               random_state=2)
```

```
X_train.sample(5)
```

	brand	km_driven	fuel	owner
2482	Hyundai	25000	Petrol	Second Owner
4079	Maruti	35000	Diesel	First Owner
6794	Tata	120000	Diesel	First Owner
2251	Hyundai	60000	Diesel	Second Owner
1488	Fiat	142500	Diesel	First Owner

```
X_test.sample(5)
```

	brand	km_driven	fuel	owner
6405	Chevrolet	80000	Diesel	First Owner
7397	Tata	80000	Diesel	First Owner
4527	Mercedes-Benz	110000	Diesel	Third Owner
7420	Tata	175000	Diesel	Second Owner
5657	Hyundai	25000	Petrol	First Owner

```
from sklearn.preprocessing import OneHotEncoder
```

```
oe=OneHotEncoder(drop='first',dtype=np.int32)
X_train_new=oe.fit_transform(X_train[['fuel','owner']]).toarray()
```

```
X_test_new=oe.transform(X_test[['fuel','owner']]).toarray()
```

```
X_test_new
```

```
array([[0, 0, 1, ..., 0, 0, 0],
       [1, 0, 0, ..., 1, 0, 0],
       [0, 0, 1, ..., 0, 0, 0],
       ...,
       [0, 0, 1, ..., 0, 0, 0],
       [0, 0, 1, ..., 1, 0, 0],
       [1, 0, 0, ..., 0, 0, 0]], dtype=int32)
```

```
X_test_new=pd.DataFrame(X_test_new)
X_train_new=pd.DataFrame(X_train_new)
```

```
X_train_new
```

	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0
3	1	0	0	0	1	0	0
4	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...
6497	1	0	0	0	0	0	0
6498	0	0	1	0	0	0	0
6499	0	0	1	0	0	0	0
6500	1	0	0	0	1	0	0
6501	1	0	0	0	0	0	0

6502 rows × 7 columns

Next steps: Generate code with `X_train_new` New interactive sheet

```
X_test_new
```

	0	1	2	3	4	5	6
0	0	0	1	0	0	0	0
1	1	0	0	0	1	0	0
2	0	0	1	0	0	0	0
3	0	0	1	0	1	0	0
4	1	0	0	0	1	0	0
...	...	...	...	...	...	...	...
1621	1	0	0	0	0	0	0
1622	0	0	1	0	0	0	0
1623	0	0	1	0	0	0	0
1624	0	0	1	0	1	0	0
1625	1	0	0	0	0	0	0

1626 rows × 7 columns