import datasets for ordinal encoding

```
import pandas as pd
import requests
from io import StringIO

url = "https://raw.githubusercontent.com/campusx-official/100-days-of-machine-learning/refs/heads/main/day26-ordinal-encoding/customer.csv"
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0"}
req = requests.get(url, headers=headers)
data = StringIO(req.text)

df = pd.read_csv(data)
```

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

	age	gender	review	education	purchased
6	18	Male	Good	School	No
28	48	Male	Poor	School	No
21	32	Male	Average	PG	No
20	57	Female	Average	School	Yes
16	59	Male	Poor	UG	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.head()
```

	review	education
33	Good	PG
35	Poor	School
26	Poor	PG
34	Average	School
18	Good	School

Next steps: Generate code with x_train New interactive sheet

OrdinalEncoder

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

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

```
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_train = pd.DataFrame(x_train , columns=['review' , 'education'])
x_train.head()
```

	review	education
0	2.0	2.0
1	0.0	0.0
2	0.0	2.0
3	1.0	0.0
4	2.0	0.0

Next steps: Generate code with x_train New interactive sheet

LabelEncoder

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y_train = le.fit_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, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0])
```

```
y_test
```

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

import a datasets for OneHotEncoding

```
import pandas as pd
import requests
from io import StringIO

url = "https://raw.githubusercontent.com/campusx-official/100-days-of-machine-learning/refs/heads/main/day27-one-hot-encoding/cars.csv"
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0"}
req = requests.get(url, headers=headers)
data = StringIO(req.text)

df = pd.read_csv(data)
```

```
df.head()
```

	brand	km_driven	fuel	owner	selling_price
0	Maruti	145500	Diesel	First Owner	450000
1	Skoda	120000	Diesel	Second Owner	370000
2	Honda	140000	Petrol	Third Owner	158000
3	Hyundai	127000	Diesel	First Owner	225000
4	Maruti	120000	Petrol	First Owner	130000

Next steps: Generate code with df New interactive sheet

```
#train_test_split
from sklearn.model_selection import  train_test_split
x_train  ,x_test , y_train,y_test = train_test_split(df.iloc[:,:4],df.iloc[:,-1],test_size=0.2 ,random_state=0)
```

```
x_train
```

	brand	km_driven	fuel	owner
3042	Hyundai	60000	LPG	First Owner
1520	Tata	150000	Diesel	Third Owner
2611	Hyundai	110000	Diesel	Second Owner
3544	Mahindra	28000	Diesel	Second Owner
4138	Maruti	15000	Petrol	First Owner
...	...	...	...	...
4931	Tata	70000	Diesel	Third Owner
3264	Ford	100000	Diesel	Second Owner
1653	Hyundai	90000	Petrol	Second Owner
2607	Volkswagen	90000	Diesel	First Owner
2732	Hyundai	110000	Petrol	First Owner

6502 rows × 4 columns

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

```
x_test
```

	brand	km_driven	fuel	owner
3558	Hyundai	40000	Diesel	First Owner
233	Mahindra	70000	Diesel	First Owner
7952	Maruti	5000	Petrol	First Owner
572	Maruti	120000	Petrol	Third Owner
6960	Lexus	20000	Petrol	First Owner
...	...	...	...	...
7576	Fiat	100000	Diesel	Third Owner
1484	Maruti	120000	Petrol	Third Owner
1881	Maruti	40000	Diesel	First Owner
4917	Hyundai	2350	Petrol	First Owner
5934	Hyundai	80000	Diesel	Second Owner

1626 rows × 4 columns

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

OneHotEncoder

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

```
ohe = OneHotEncoder(drop='first')
x_train_new = ohe.fit_transform(x_train[['fuel','owner']]).toarray()
x_test_new = ohe.transform(x_test[['fuel','owner']]).toarray()
```

```
x_train_new
```

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

```
x_train_new = pd.DataFrame(x_test_new)
x_train_new
```

	0	1	2	3	4	5	6
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	0.0	0.0	0.0	1.0
4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...
1621	1.0	0.0	0.0	0.0	0.0	0.0	1.0
1622	0.0	0.0	1.0	0.0	0.0	0.0	1.0
1623	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1624	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1625	1.0	0.0	0.0	0.0	1.0	0.0	0.0

1626 rows × 7 columns

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