```
import pandas as pd
```

```
cars = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/MPG.csv')
```

```
cars.head(10)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origi
0	18.0	8	307.0	130.0	3504	12.0	70	us
1	15.0	8	350.0	165.0	3693	11.5	70	us
2	18.0	8	318.0	150.0	3436	11.0	70	us
3	16.0	8	304.0	150.0	3433	12.0	70	us
4	17.0	8	302.0	140.0	3449	10.5	70	us
5	15.0	8	429.0	198.0	4341	10.0	70	us

```
cars.tail()
```

```
pd.options.display.max_rows = None
```

```
cars
```

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Boston.csv')
```

```
df.head()
```

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	I
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	

```
df.shape
```

```
(506, 14)
```

```
df.columns
```

```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64  
 4   NX       506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
```

```
dtypes: float64(12), int64(2)
```

```
df.describe()
```

```
y = df['MEDV']
```

```
y.shape
```

```
(506,)
```

```
X = df[['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO',
```

```
X.shape
```

```
(506, 13)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state =
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((354, 13), (152, 13), (354,), (152,))
```

```
from sklearn.preprocessing import StandardScaler
```

```
ss = StandardScaler()
```

```
X_train_ss = ss.fit_transform(X_train)
```

```
X_test_ss = ss.fit_transform(X_test)
```

```
X_train_ss
```

```
array([[-0.14113619, -0.48175769, -0.19860022, ...,  0.00438903,
        -0.05084503, -0.01555641],
       [-0.42121529,  3.02166196, -1.33410259, ..., -1.68641979,
         0.42969249, -1.33650784],
       [-0.41266839, -0.48175769,  0.22414717, ...,  0.14148164,
         0.19739169, -0.10842497],
       ...,
       [-0.38944304, -0.48175769, -0.19860022, ...,  0.00438903,
         0.37963873,  0.77313338],
       [-0.41404001,  0.41002186, -0.81324318, ..., -0.72677154,
         0.43161763,  0.09671754],
       [-0.41578561,  2.06618387, -1.3831586 , ..., -0.04130851,
         0.39707198, -0.68781395]])
```

```
X_test_ss
```

```
array([[-0.36714008, -0.50235603, -0.6925381 , ..., -0.57641511,
         0.2366856 , -1.24860568],
       [-0.40880876, -0.50235603, -0.58591169, ..., -0.33768188,
         0.43031542, -0.31886558],
       [-0.41291768, -0.50235603, -0.12035979, ..., -0.38542852,
         0.36717526,  0.17122998],
       ...,
       [-0.34428827, -0.50235603,  1.66375525, ...,  1.23795746,
         0.30005961, -0.18769294],
       [-0.05769974, -0.50235603,  1.31684399, ..., -1.86557456,
        -0.3514533 , -0.15886379],
       [-0.42293258,  1.25907688, -0.66100071, ..., -0.48092181,
         0.43031542, -0.75418575]])
```

```
X_train_ss.mean(axis = 0)
```

```
array([ 7.52693576e-18,  2.50897859e-17,  5.01795717e-17,  1.12904036e-17,
       -4.74196953e-16, -1.03369918e-15, -1.85664415e-16,  7.27603790e-17,
       -3.51257002e-17,  1.15413015e-16,  7.32621747e-16, -4.01436574e-17,
       -3.51257002e-17])
```

```
X_test_ss.mean(axis = 0)
```

```
array([-1.75298372e-17,  3.35988547e-17, -1.16865582e-17,  1.75298372e-17,
       -1.27091320e-16, -4.26559373e-16,  2.62947558e-17,  2.07436407e-16,
       -5.25895117e-17, -1.16865582e-17,  3.15537070e-16, -7.24566606e-16,
       -1.31473779e-16])
```

```
X_train_ss.std(axis = 0)
```

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

```
X_test_ss.std(axis = 0)
```

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

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Customer%20Purchase.
```

```
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.shape
```

```
(50, 6)
```

df.columns

```
df.columns
```

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

```
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
```

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

X

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

```
oe = OrdinalEncoder()
```

```
X = oe.fit_transform(X)
```

```
X
```

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

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

```
oe.categories_
```

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

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

```
array([['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)
```

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

```
oe = OrdinalEncoder(categories=[['Male', 'Female'],['School','UG','PG'],['Poor', 'Average'
```

```
x = oe.fit_transform(X)
```

```
oe.categories_
```

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

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Online%20Purchase.cs
df.head()
```

	Customer_ID	Gender	Age	Salary	Purchased
0	1	Male	35	500	0
1	2	Female	25	300000	1
2	3	Female	100	200000	0
3	15566689	Female	35	57000	0
4	15569641	Female	58	95000	1

```
df.shape
```

```
(403, 5)
```

```
df.columns
```

```
Index(['Customer_ID', 'Gender', 'Age', 'Salary', 'Purchased'], dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 403 entries, 0 to 402
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Customer_ID  403 non-null    int64
 1   Gender       403 non-null    object
 2   Age          403 non-null    int64
 3   Salary       403 non-null    int64
 4   Purchased    403 non-null    int64
dtypes: int64(4), object(1)
memory usage: 15.9+ KB
```

```
df.describe()
```

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

```
(403,)
```

```
X = df[['Age','Salary']]
```

```
X.shape
```

```
(403, 2)
```

```
df[['Age']].hist();
```

```
df[['Age']].boxplot()
```

```
df[['Salary']].hist();
```

```
df[['Salary']].boxplot()
```

```
plt.scatter(df['Age'],df['Salary']);
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, stratify = y, r

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((282, 2), (121, 2), (282,), (121,))
```

```
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()

X_train_ss = ss.fit_transform(X_train)

X_test_ss = ss.fit_transform(X_test)

X_train_ss = pd.DataFrame(X_train_ss, columns=X_train.columns)
X_test_ss = pd.DataFrame(X_test_ss, columns=X_test.columns)

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

ax1.scatter(X_train['Age'], X_train['Salary'])
ax1.set_title("Before Scaling")
ax2.scatter(X_train_ss['Age'], X_train_ss['Salary'],color='red')
ax2.set_title("After Scaling")
plt.show()
```

```
2fig, (ax1, ax) = plt.subplots(ncols=2, figsize=(12, 5))
ax1.set_title('Before Scaling')
sns.kdeplot(X_train['Age'], ax=ax1)
```

```
sns.kdeplot(X_train['Salary'], ax=ax1)


ax1.set_title('After Standard Scaling')
sns.kdeplot(X_train_ss['Age'], ax=ax2)
sns.kdeplot(X_train_ss['Salary'], ax=ax2)
plt.show()
```

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))
ax1.set_title('Age Distribution Before Scaling')
sns.kdeplot(X_train_ss['Age'], ax=ax1)

ax2.set_title('Age Distribution Before Scaling')
sns.kdeplot(X_train_ss['Age'], ax=ax2)
plt.show()
```

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))
ax1.set_title('Salary Distribution Before Scaling')
sns.kdeplot(X_train_ss['Salary'], ax=ax1)

ax2.set_title('Salary Distribution Before Scaling')
sns.kdeplot(X_train_ss['Salary'], ax=ax2)
plt.show()
```

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()

lr.fit(X_train, y_train)
```

```
     LogisticRegression()
```

```
y_pred = lr.predict(X_test)

from sklearn.metrics import accuracy_score

accuracy_score(y_test, y_pred)
```

```
     0.6446280991735537
```

```
lr.fit(X_train_ss, y_train)
```

```
     LogisticRegression()
```

```
y_pred = lr.predict(X_test_ss)
```

```
accuracy_score(y_test, y_pred)
```

```
0.8099173553719008
```

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Customer%20Purchase.
```

```
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.shape
```

```
(50, 6)
```

```
df.columns
```

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

```
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
```

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

```
y
```

```
0      No
1      No
```

```
2      No
3      No
4      No
5     Yes
6      No
7     Yes
8      No
9     Yes
10    Yes
11    Yes
12     No
13     No
14    Yes
15     No
16    Yes
17    Yes
18     No
19    Yes
20    Yes
21     No
22    Yes
23     No
24    Yes
25     No
26     No
27     No
28     No
29    Yes
30     No
31    Yes
32    Yes
33    Yes
34     No
35    Yes
36    Yes
37    Yes
38     No
39     No
40     No
41    Yes
42    Yes
43     No
44     No
45    Yes
46     No
47    Yes
48    Yes
49     No
Name: Purchased, dtype: object
```

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

```
le = LabelEncoder()
```

```
y = le.fit_transform(y)
```

```
y
```

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

```
le.classes_
```

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

```
le.inverse_transform([0, 0, 1, 1])
```

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

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r'https://github.com/YBI-Foundation/Dataset/raw/main/Customer%20Purchase.
```

```
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.shape
```

```
(50, 6)
```

```
df.columns
```

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

```
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
```

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

```
X
```

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

```
ohe = OneHotEncoder()
```

```
X_Gender = ohe.fit_transform(X[['Gender']])
```

```
ohe.categories_
```

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

```
X_Gender.toarray().shape
```

```
(50, 2)
```

```
X_Gender.toarray()
```

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

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

```
X_Education = ohe.fit_transform(X[['Education']])
```

```
ohe.categories_
```

```
[array(['PG', 'School', 'UG'], dtype=object)]
```

```
X_Education.toarray().shape
```

```
(50, 3)
```

```
X_Education.toarray()
```

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

```
X_Review = ohe.fit_transform(X[['Review']])
```

```
ohe.categories_
```

```
[array(['Average', 'Good', 'Poor'], dtype=object)]
```

```
X_Review.toarray().shape
```

```
(50, 3)
```

```
X_Review.toarray()
```

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

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

```
ohe = OneHotEncoder(categories=[['Male', 'Female'],['School','UG','PG'],['Poor', 'Average'
```

```
X = ohe.fit_transform(X)
ohe.categories_
```

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

```
X.toarray().shape
```

```
(50, 8)
```

```
X.toarray()
```

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

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

```
ohe.inverse_transform([[0., 1., 1., 0., 0., 0., 1., 0.]])
```

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

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

```
ohe = OneHotEncoder(drop='first')
```

```
X = ohe.fit_transform(X)
```

```
ohe.categories_
```

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

```
X.toarray().shape
```

```
(50, 5)
```

```
X.toarray()
```

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

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