In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [10]:

```
df=pd.read_csv('datasets/black_friday.csv')
df
```

Out[10]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	
550065	1006036	P00375436	F	26-35	15	B	
550066	1006038	P00375436	F	55+	1	C	
550067	1006039	P00371644	F	46-50	0	B	

550068 rows × 12 columns

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     550068 non-null  int64  
 1   Product_ID                  550068 non-null  object 
 2   Gender                      550068 non-null  object 
 3   Age                         550068 non-null  object 
 4   Occupation                  550068 non-null  int64  
 5   City_Category               550068 non-null  object 
 6   Stay_In_Current_City_Years  550068 non-null  object 
 7   Marital_Status              550068 non-null  int64  
 8   Product_Category_1          550068 non-null  int64  
 9   Product_Category_2          376430 non-null  float64
 10  Product_Category_3          166821 non-null  float64
 11  Purchase                    550068 non-null  int64  
dtypes: float64(2), int64(5), object(5)
memory usage: 50.4+ MB
```

In [5]:

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

Out[5]:

```
User_ID                            0
Product_ID                         0
Gender                             0
Age                                0
Occupation                         0
City_Category                      0
Stay_In_Current_City_Years         0
Marital_Status                     0
Product_Category_1                 0
Product_Category_2            173638
Product_Category_3            383247
Purchase                           0
dtype: int64
```

In [12]:

```
df['Product_Category_2'].value_counts()
```

Out[12]:

```
8.0     64088
14.0    55108
2.0     49217
16.0    43255
15.0    37855
5.0     26235
4.0     25677
6.0     16466
11.0    14134
17.0    13320
13.0    10531
9.0      5693
12.0     5528
10.0     3043
3.0      2884
18.0     2770
7.0       626
Name: Product_Category_2, dtype: int64
```

In [14]:

```
df.dropna(axis=1,inplace=True)
df
```

Out[14]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	
550065	1006036	P00375436	F	26-35	15	B	
550066	1006038	P00375436	F	55+	1	C	
550067	1006039	P00371644	F	46-50	0	B	

550068 rows × 10 columns

In [15]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype 
---  ------                      --------------   ----- 
 0   User_ID                     550068 non-null  int64 
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64 
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64 
 8   Product_Category_1          550068 non-null  int64 
 9   Purchase                    550068 non-null  int64 
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

```
In [16]:
```

```
df=pd.read_csv('datasets/black_friday.csv')
df
```

```
Out[16]:
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	
550065	1006036	P00375436	F	26-35	15	B	
550066	1006038	P00375436	F	55+	1	C	
550067	1006039	P00371644	F	46-50	0	B	

550068 rows × 12 columns

In [17]:

```
df.dropna(axis=0,inplace=True)
df
```

Out[17]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
1	1000001	P00248942	F	0-17	10	A	
6	1000004	P00184942	M	46-50	7	B	
13	1000005	P00145042	M	26-35	20	A	
14	1000006	P00231342	F	51-55	9	A	
16	1000006	P0096642	F	51-55	9	A	
...	...	...	...	...	...	...	
545902	1006039	P00064042	F	46-50	0	B	
545904	1006040	P00081142	M	26-35	6	B	
545907	1006040	P00277642	M	26-35	6	B	
545908	1006040	P00127642	M	26-35	6	B	
545914	1006040	P00217442	M	26-35	6	B	

166821 rows × 12 columns

In [18]:

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

Out[18]:

```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category_1            0
Product_Category_2            0
Product_Category_3            0
Purchase                      0
dtype: int64
```

In [19]:

```
df = pd.read_csv('datasets/black_friday.csv')
df.isnull().sum()
```

Out[19]:

```
User_ID                            0
Product_ID                         0
Gender                             0
Age                                0
Occupation                         0
City_Category                      0
Stay_In_Current_City_Years         0
Marital_Status                     0
Product_Category_1                 0
Product_Category_2            173638
Product_Category_3            383247
Purchase                           0
dtype: int64
```

In [20]:

```
df['Product_Category_2'].fillna(df['Product_Category_2'].mean(),inplace=True)
df['Product_Category_3'].fillna(df['Product_Category_3'].mean(),inplace=True)
```

In [21]:

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

Out[21]:

```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category_1            0
Product_Category_2            0
Product_Category_3            0
Purchase                      0
dtype: int64
```

In [22]:

```
df.describe()
```

Out[22]:

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_2	F
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000	
mean	1.003029e+06	8.076707	0.409653	5.404270	9.842329	
std	1.727592e+03	6.522660	0.491770	3.936211	4.207852	
min	1.000001e+06	0.000000	0.000000	1.000000	2.000000	
25%	1.001516e+06	2.000000	0.000000	1.000000	8.000000	
50%	1.003077e+06	7.000000	0.000000	5.000000	9.842329	
75%	1.004478e+06	14.000000	1.000000	8.000000	14.000000	
max	1.006040e+06	20.000000	1.000000	20.000000	18.000000	

In [23]:

```
df
```

Out[23]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	
550065	1006036	P00375436	F	26-35	15	B	
550066	1006038	P00375436	F	55+	1	C	
550067	1006039	P00371644	F	46-50	0	B	

550068 rows × 12 columns

In [26]:

```
df=pd.get_dummies(df,columns=['Age'],drop_first=True)
df
```

Out[26]:

	User_ID	Product_ID	Gender	Occupation	City_Category	Product_Category_1	Product
0	1000001	P00069042	F	10	A	3	
1	1000001	P00248942	F	10	A	1	
2	1000001	P00087842	F	10	A	12	
3	1000001	P00085442	F	10	A	12	
4	1000002	P00285442	M	16	C	8	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	13	B	20	
550064	1006035	P00375436	F	1	C	20	
550065	1006036	P00375436	F	15	B	20	
550066	1006038	P00375436	F	1	C	20	
550067	1006039	P00371644	F	0	B	20	

550068 rows × 20 columns

In [27]:

```
df.describe()
```

Out[27]:

	User_ID	Occupation	Product_Category_1	Product_Category_2	Product_Categor
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.0000
mean	1.003029e+06	8.076707	5.404270	9.842329	12.668
std	1.727592e+03	6.522660	3.936211	4.207852	2.271
min	1.000001e+06	0.000000	1.000000	2.000000	3.0000
25%	1.001516e+06	2.000000	1.000000	8.000000	12.668
50%	1.003077e+06	7.000000	5.000000	9.842329	12.668
75%	1.004478e+06	14.000000	8.000000	14.000000	12.668
max	1.006040e+06	20.000000	20.000000	18.000000	18.0000

In [29]:

```
x=df.drop('Purchase',axis=1)
y=df['Purchase']
```

In [30]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
x_train
```

Out[30]:

	User_ID	Product_ID	Gender	Occupation	City_Category	Product_Category_1	Product
405592	1002384	P00003442	M	17	C	4	
508755	1000352	P00259342	M	4	A	5	
528657	1003477	P0096842	F	1	A	3	
122241	1000891	P00276642	M	1	C	8	
98700	1003311	P00249842	M	4	A	8	
...	...	...	...	...	...	...	
229488	1005394	P00058042	M	0	A	8	
232976	1005924	P00010742	M	0	B	1	
238394	1000796	P00200242	M	6	B	8	
347859	1005576	P00191642	M	4	C	3	
368805	1002837	P00234542	M	0	B	5	

440054 rows × 19 columns

In [31]:

```
x_test
```

Out[31]:

	User_ID	Product_ID	Gender	Occupation	City_Category	Product_Category_1	Product
181825	1004050	P00044442	M	6	B	1	
284309	1001764	P0097842	M	0	B	5	
173933	1002913	P00209842	F	20	B	5	
171386	1002476	P00112142	M	12	C	1	
488500	1003315	P00363942	M	12	B	5	
...	...	...	...	...	...	...	
148936	1004979	P00283042	M	2	B	5	
479499	1001835	P00196142	M	19	B	3	
313312	1000276	P00026042	M	16	C	8	
525805	1003032	P00272242	M	0	A	8	
218315	1003679	P00321342	M	4	A	1	

110014 rows × 19 columns

In [33]:

```
y_train.value_counts()
```

Out[33]:

```
7193     156
7011     156
7027     155
6855     155
6960     154
        ...
22984      1
18562      1
14640      1
343        1
14070      1
Name: Purchase, Length: 17670, dtype: int64
```

In [34]:

```
df
```

Out[34]:

	User_ID	Product_ID	Gender	Occupation	City_Category	Product_Category_1	Product
0	1000001	P00069042	F	10	A	3	
1	1000001	P00248942	F	10	A	1	
2	1000001	P00087842	F	10	A	12	
3	1000001	P00085442	F	10	A	12	
4	1000002	P00285442	M	16	C	8	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	13	B	20	
550064	1006035	P00375436	F	1	C	20	
550065	1006036	P00375436	F	15	B	20	
550066	1006038	P00375436	F	1	C	20	
550067	1006039	P00371644	F	0	B	20	

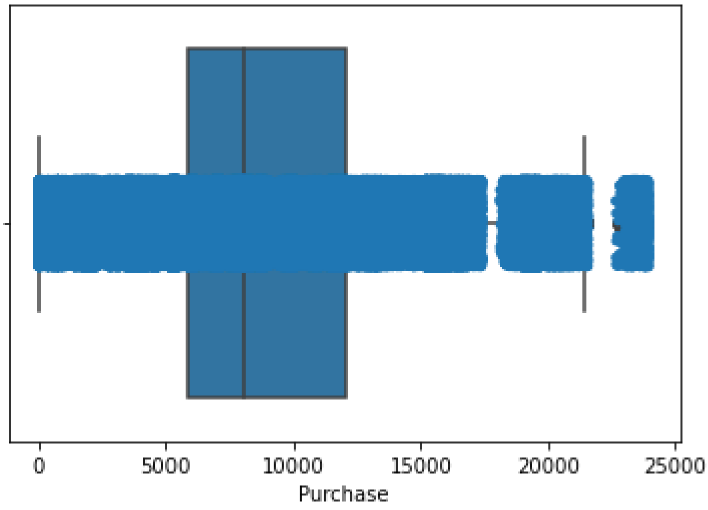
550068 rows × 20 columns

In [36]:

```
sns.boxplot(x='Purchase',data=df)
sns.stripplot(x='Purchase',data=df)
```

Out[36]:

```
<AxesSubplot:xlabel='Purchase'>
```



In [38]:

```
from datasist.structdata import detect_outliers
outlier_indices=detect_outliers(df,0,['Purchase'])
outlier_indices
```

Out[38]:

```
[343,
 375,
 652,
 736,
 1041,
 1445,
 1902,
 3166,
 3167,
 3172,
 3391,
 3630,
 3908,
 4148,
 4221,
 4527,
 5059,
 5060,
```

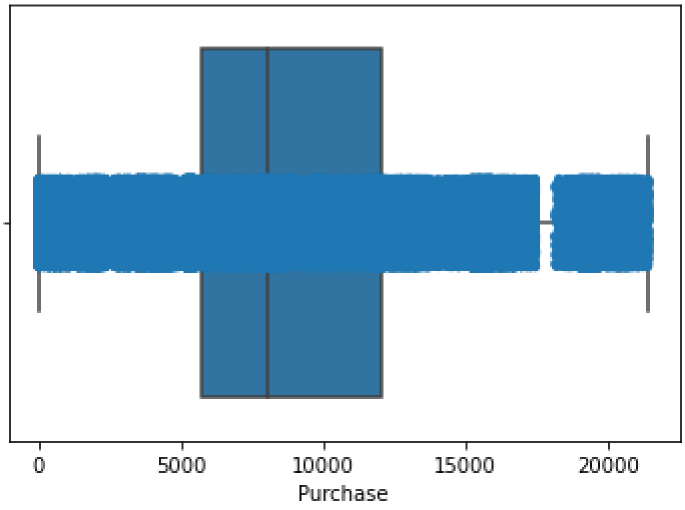
In [39]:

```
df.drop(outlier_indices,inplace=True)

sns.boxplot(x='Purchase',data=df)
sns.stripplot(x='Purchase',data=df)
```

Out[39]:

```
<AxesSubplot:xlabel='Purchase'>
```



In [44]:

```
df
```

Out[44]:

	User_ID	Product_ID	Gender	Occupation	City_Category	Product_Category_1	Product
0	1000001	P00069042	F	10	A	3	
1	1000001	P00248942	F	10	A	1	
2	1000001	P00087842	F	10	A	12	
3	1000001	P00085442	F	10	A	12	
4	1000002	P00285442	M	16	C	8	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	13	B	20	
550064	1006035	P00375436	F	1	C	20	
550065	1006036	P00375436	F	15	B	20	
550066	1006038	P00375436	F	1	C	20	
550067	1006039	P00371644	F	0	B	20	

547391 rows × 20 columns

In [45]:

```
x=df.drop('Occupation',axis=1)
y=df['Occupation']
```

In [46]:

```
y.value_counts()
```

Out[46]:

```
4     72040
0     69310
7     58875
1     47174
17    39855
20    33355
12    30995
14    27173
2     26435
16    25251
6     20261
3     17568
10    12888
5     12133
15    12086
11    11500
19     8412
13     7667
18     6595
9      6278
8      1540
Name: Occupation, dtype: int64
```

In [47]:

```
(y.value_counts()/len(df))*100
```

Out[47]:

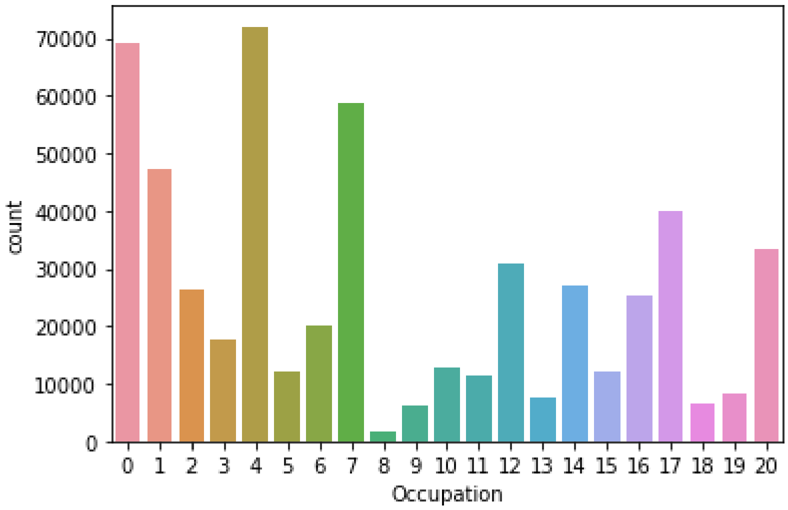
```
4     13.160611
0     12.661882
7     10.755566
1      8.617971
17     7.280902
20     6.093451
12     5.662315
14     4.964093
2      4.829272
16     4.612973
6      3.701376
3      3.209406
10     2.354441
5      2.216514
15     2.207928
11     2.100875
19     1.536744
13     1.400644
18     1.204806
9      1.146895
8      0.281335
Name: Occupation, dtype: float64
```

In [48]:

```
sns.countplot(x='Occupation',data=df)
```

Out[48]:

```
<AxesSubplot:xlabel='Occupation', ylabel='count'>
```



In [49]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
y_train.value_counts()
```

Out[49]:

```
4     57657
0     55594
7     47184
1     37706
17    31783
20    26595
12    24825
14    21740
2     21145
16    20199
6     16220
3     13977
10    10359
15     9705
5      9613
11     9253
19     6729
13     6100
18     5259
9      5010
8      1259
Name: Occupation, dtype: int64
```

In [54]:

```
x=df[['Occupation','Product_Category_3','Product_Category_1']]
y=df['Purchase']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
x_train
```

Out[54]:

	Occupation	Product_Category_3	Product_Category_1
424732	0	12.668243	8
244450	14	12.668243	8
541457	11	12.668243	8
4943	16	4.000000	2
485856	11	12.668243	8
...	...	...	...
362319	0	12.668243	8
533336	3	16.000000	1
98218	4	12.668243	8
299030	6	12.668243	16
64256	16	12.668243	5

410543 rows × 3 columns

In [55]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
x_train
```

Out[55]:

```
array([[0.        , 0.64454955, 0.36842105],
       [0.7       , 0.64454955, 0.36842105],
       [0.55      , 0.64454955, 0.36842105],
       ...,
       [0.2       , 0.64454955, 0.36842105],
       [0.3       , 0.64454955, 0.78947368],
       [0.8       , 0.64454955, 0.21052632]])
```

In [56]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
x_train
```

Out[56]:

	Occupation	Product_Category_3	Product_Category_1
475261	15	14.000000	5
187985	7	12.668243	15
487431	4	12.668243	8
120142	8	12.668243	1
286987	0	12.668243	1
...	...	...	...
104457	14	12.668243	5
113147	12	12.668243	8
346900	1	12.668243	8
207487	9	12.668243	8
178652	4	12.668243	5

410543 rows × 3 columns

In [57]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train=scaler.fit_transform(x_train)
x_test=scaler.transform(x_test)
x_train
```

Out[57]:

```
array([[ 1.06119953,  0.58795959, -0.09670154],
       [-0.16515434,  0.00285399,  2.44777138],
       [-0.62503704,  0.00285399,  0.66664034],
       ...,
       [-1.08491975,  0.00285399,  0.66664034],
       [ 0.14143412,  0.00285399,  0.66664034],
       [-0.62503704,  0.00285399, -0.09670154]])
```

In [ ]:

```

```