EDA on Black Friday Data

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

Loading the data

```
df = pd.read_csv(r"C:\Users\Lokesh\Downloads\CA 2\CA 2\
blackfriday.csv")
df
```

```
        User_ID Product_ID Gender    Age  Occupation City_Category  \
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

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
0                               2               0                   3

1                               2               0                   1

2                               2               0                  12

3                               2               0                  12

4                              4+               0                   8

...                           ...             ...                 ...

550063                          1               1                  20

550064                          3               0                  20

550065                         4+               1                  20

550066                          2               0                  20
```

```
550067                          4+               1                  20


        Product_Category_2  Product_Category_3  Purchase
0                      NaN                 NaN      8370
1                      6.0                14.0     15200
2                      NaN                 NaN      1422
3                     14.0                 NaN      1057
4                      NaN                 NaN      7969
...                    ...                 ...       ...
550063                 NaN                 NaN       368
550064                 NaN                 NaN       371
550065                 NaN                 NaN       137
550066                 NaN                 NaN       365
550067                 NaN                 NaN       490

[550068 rows x 12 columns]
```

```
df.head()
```

```
   User_ID Product_ID Gender   Age  Occupation City_Category  \
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

  Stay_In_Current_City_Years  Marital_Status  Product_Category_1  \
0                          2               0                   3
1                          2               0                   1
2                          2               0                  12
3                          2               0                  12
4                         4+               0                   8

   Product_Category_2  Product_Category_3  Purchase
0                 NaN                 NaN      8370
1                 6.0                14.0     15200
2                 NaN                 NaN      1422
3                14.0                 NaN      1057
4                 NaN                 NaN      7969
```

Checking Columns and there datatypes

```
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
```

```
df[['User_ID','Marital_Status']].tail()
```

```
        User_ID  Marital_Status
550063  1006033               1
550064  1006035               0
550065  1006036               1
550066  1006038               0
550067  1006039               1
```

Checking the Null Values

```
df.isnull().mean()
```

```
User_ID                       0.000000
Product_ID                    0.000000
Gender                        0.000000
Age                           0.000000
Occupation                    0.000000
City_Category                 0.000000
Stay_In_Current_City_Years    0.000000
Marital_Status                0.000000
Product_Category_1            0.000000
Product_Category_2            0.315666
Product_Category_3            0.696727
Purchase                      0.000000
dtype: float64
```

Finding the Unique values for both columns which contaians Null values .

```
df["Product_Category_2"].unique()
```

```
array([nan,  6., 14.,  2.,  8., 15., 16., 11.,  5.,  3.,  4., 12.,
9.,
       10., 17., 13.,  7., 18.])
```

```
df["Product_Category_3"].unique()
```

```
array([nan, 14., 17.,  5.,  4., 16., 15.,  8.,  9., 13.,  6., 12.,
3.,
       18., 11., 10.])
```

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

```
9.0
```

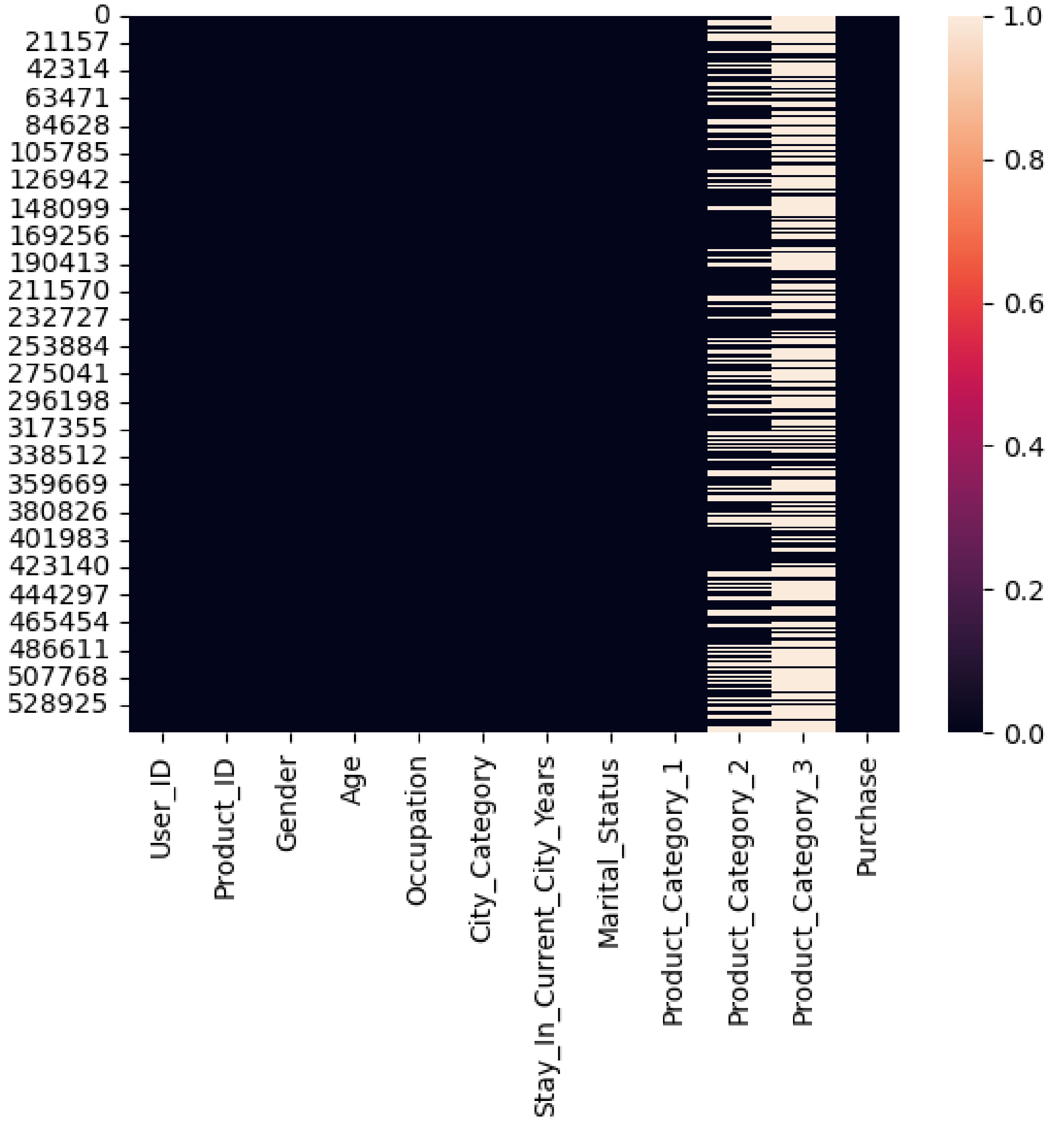
```
df['Product_Category_3'].median()
```

```
14.0
```

Checking the Null Values again with the help of Heat Map

```
sns.heatmap(df.isnull())
```

```
<Axes: >
```


0
21157
42314
63471
84628
105785
126942
148099
169256
190413
211570
232727
253884
275041
296198
317355
338512
359669
380826
401983
423140
444297
465454
486611
507768
528925
User_ID
Product_ID
Gender
Age
Occupation
City_Category
Stay_In_Current_City_Years
Marital_Status
Product_Category_1
Product_Category_2
Product_Category_3
Purchase
1.0
0.8
0.6
0.4
0.2
0.0

Handling Null Values

Product_Category_2

Step1- To Handle the null values in Product_Category_2 , I will replace the null values with the Mean.

Step2- Data Type is in Float for this column, So after replacing the null values with the Mean, I will change the data type from Float to Int.

```
df.Product_Category_2.value_counts()
```

```
Product_Category_2
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: count, dtype: int64
```

```
df.Product_Category_2.describe()
```

```
count    376430.000000
mean          9.842329
std           5.086590
min           2.000000
25%           5.000000
50%           9.000000
75%          15.000000
max          18.000000
Name: Product_Category_2, dtype: float64
```

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

```
9.842329251122386

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

df

        User_ID Product_ID Gender    Age  Occupation City_Category  \
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

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
0                               2               0                   3

1                               2               0                   1

2                               2               0                  12

3                               2               0                  12

4                              4+               0                   8

...                           ...             ...                 ...

550063                          1               1                  20

550064                          3               0                  20

550065                         4+               1                  20

550066                          2               0                  20

550067                         4+               1                  20


        Product_Category_2  Product_Category_3  Purchase
0                 9.842329                 NaN      8370
1                 6.000000                14.0     15200
2                 9.842329                 NaN      1422
3                14.000000                 NaN      1057
4                 9.842329                 NaN      7969
```

```
...                   ...                   ...        ...
550063           9.842329                   NaN        368
550064           9.842329                   NaN        371
550065           9.842329                   NaN        137
550066           9.842329                   NaN        365
550067           9.842329                   NaN        490

[550068 rows x 12 columns]
```

Product_Category_3

Step1- To Handle the null values in Product_Category_3 , I will replace the null values with the Zero (0).

Step2- Data Type is in Float for this column also, So after replacing the null values with the Zero (0), I will change the data type from Float to Int.

```
df["Product_Category_3"].fillna(0, inplace=True)

df

        User_ID Product_ID Gender    Age  Occupation City_Category  \
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

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
0                               2               0                   3

1                               2               0                   1

2                               2               0                  12

3                               2               0                  12

4                              4+               0                   8
```

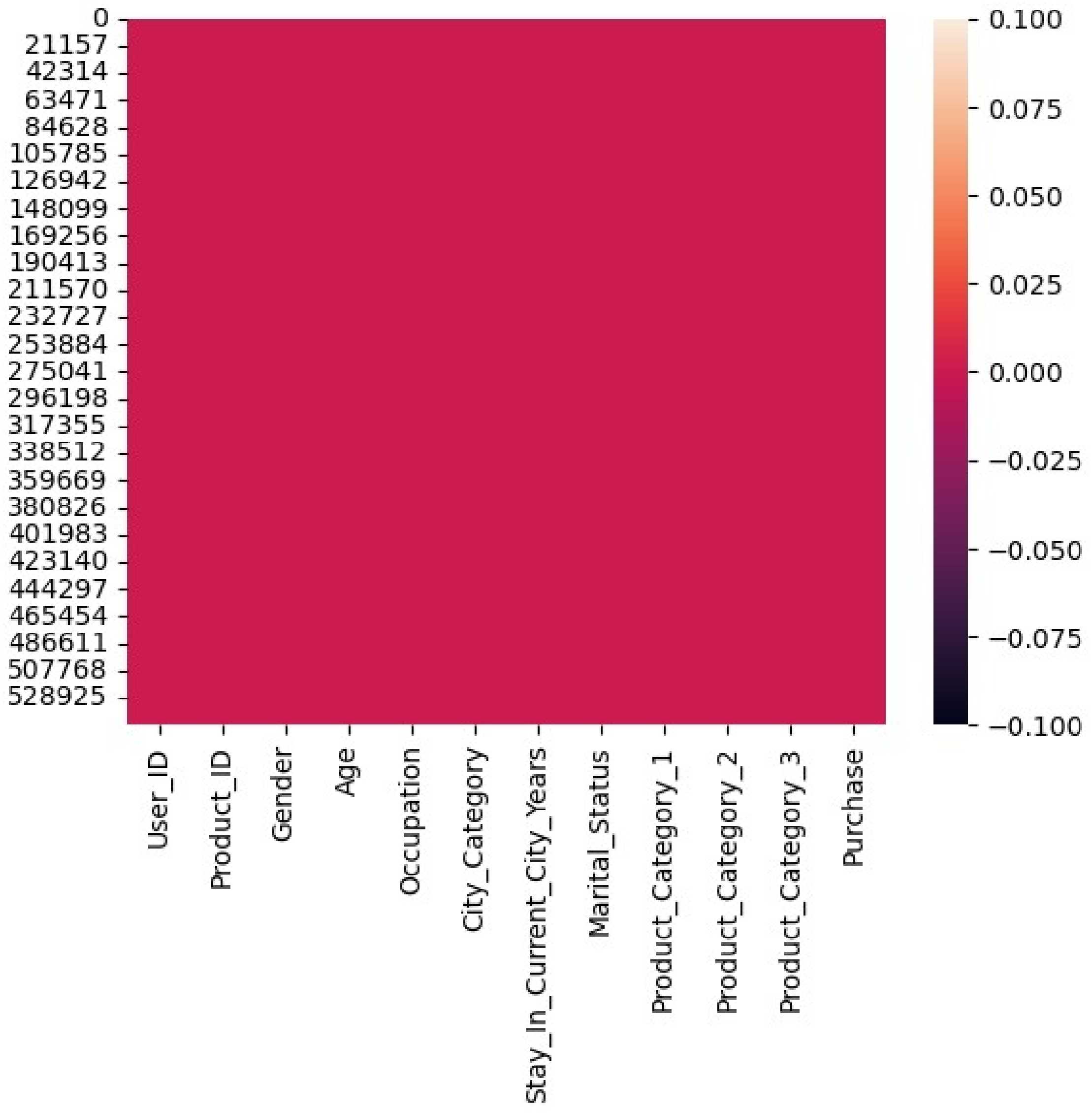
```
...                           ...             ...                 ...
550063                          1               1                  20
550064                          3               0                  20
550065                         4+               1                  20
550066                          2               0                  20
550067                         4+               1                  20

        Product_Category_2  Product_Category_3  Purchase
0                 9.842329                 0.0      8370
1                 6.000000                14.0     15200
2                 9.842329                 0.0      1422
3                14.000000                 0.0      1057
4                 9.842329                 0.0      7969
...                    ...                 ...       ...
550063            9.842329                 0.0       368
550064            9.842329                 0.0       371
550065            9.842329                 0.0       137
550066            9.842329                 0.0       365
550067            9.842329                 0.0       490

[550068 rows x 12 columns]
```

```
## Changing the Data Types

df['Product_Category_2'] = df['Product_Category_2'].astype(int)
df['Product_Category_3'] = df['Product_Category_3'].astype(int)

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          550068 non-null  int32
 10  Product_Category_3          550068 non-null  int32
 11  Purchase                    550068 non-null  int64
dtypes: int32(2), int64(5), object(5)
memory usage: 46.2+ MB
```

```
sns.heatmap(df.isnull())
```

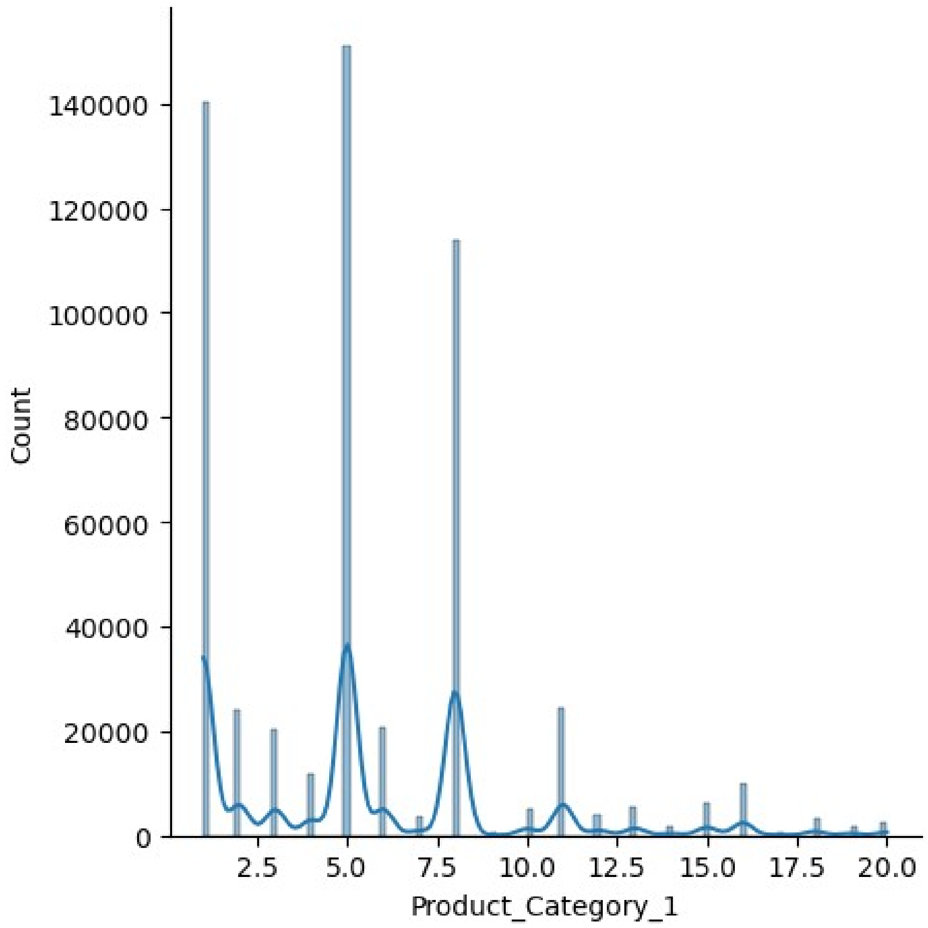
```
<Axes: >
```



```
sns.displot(df['Product_Category_1'],kde= True)
```

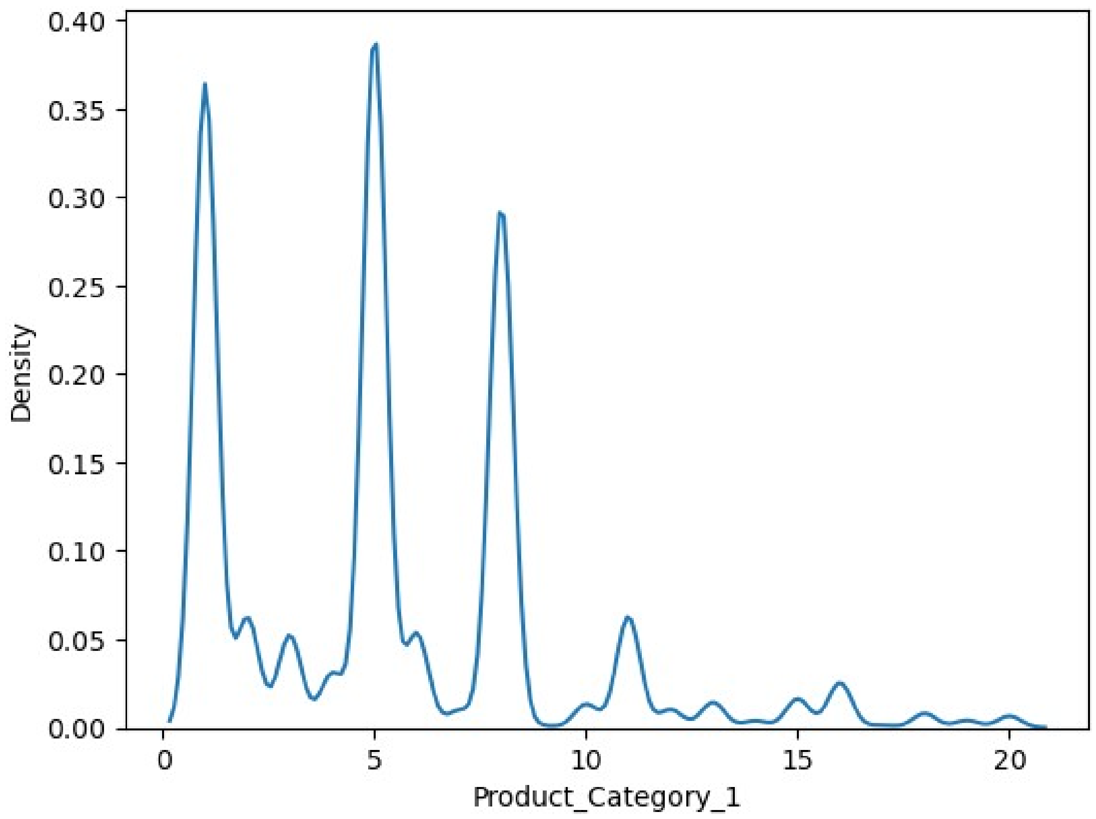
```
C:\Users\Lokesh\Python\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
<seaborn.axisgrid.FacetGrid at 0x23e9b595050>
```



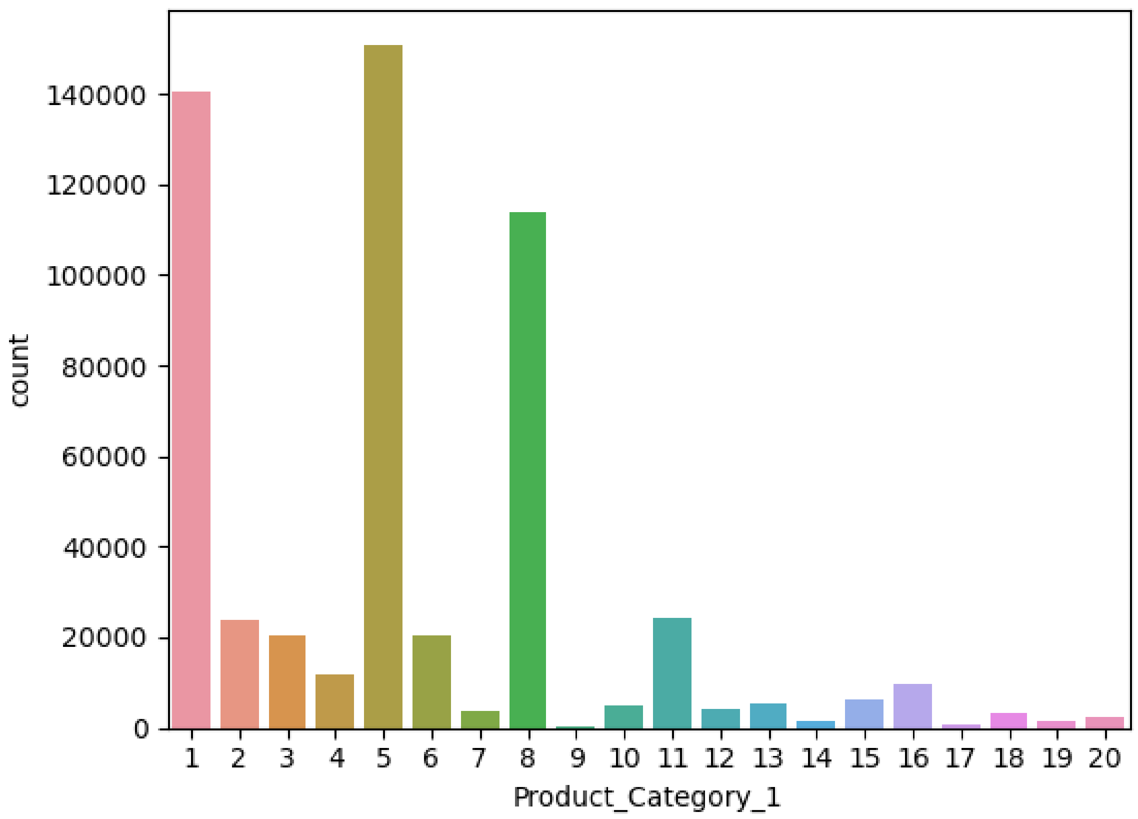
```
sns.kdeplot(df['Product_Category_1'])
```

```
<Axes: xlabel='Product_Category_1', ylabel='Density'>
```



```
sns.countplot(x='Product_Category_1',data=df)
```

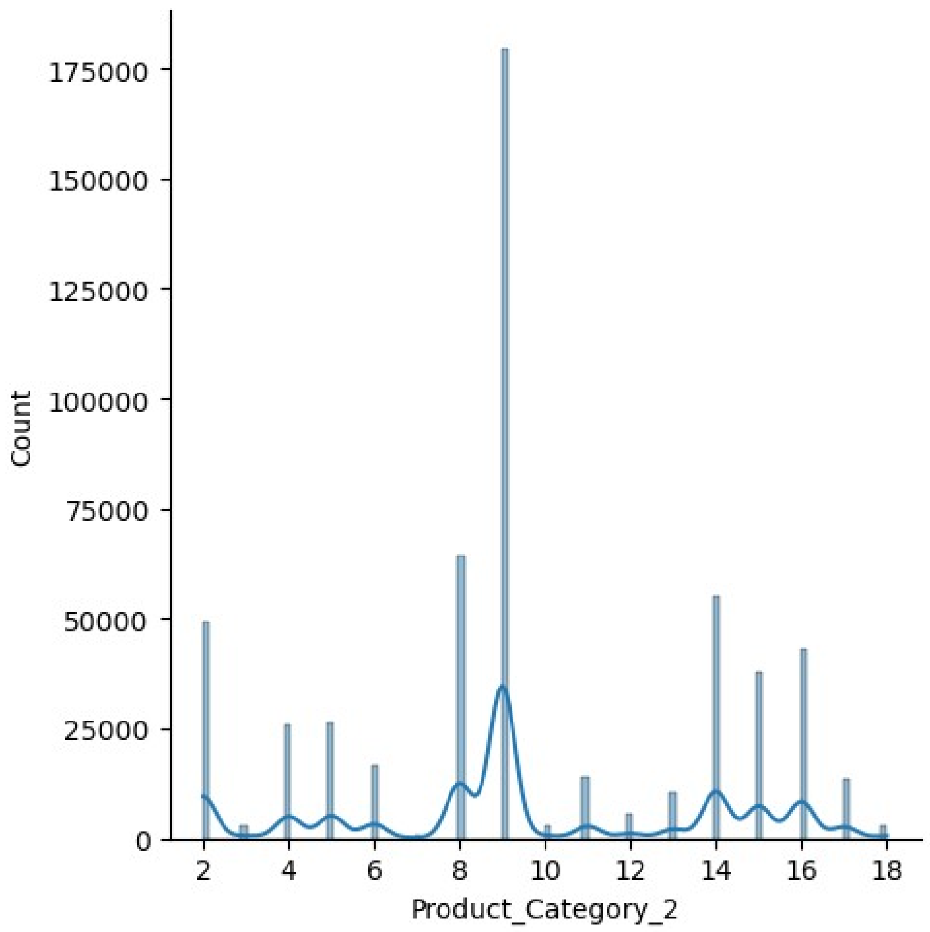
```
<Axes: xlabel='Product_Category_1', ylabel='count'>
```



```
sns.displot(df['Product_Category_2'],kde= True)
```

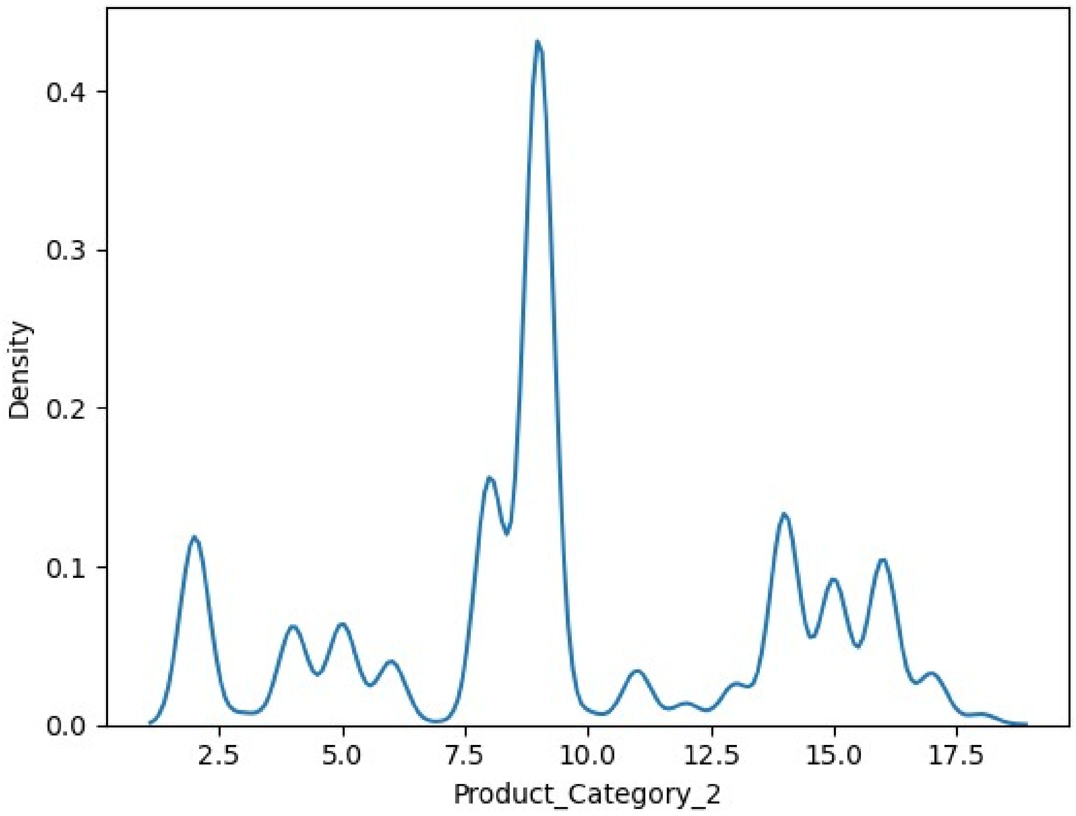
```
C:\Users\Lokesh\Python\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
<seaborn.axisgrid.FacetGrid at 0x23ea1eaaf50>
```



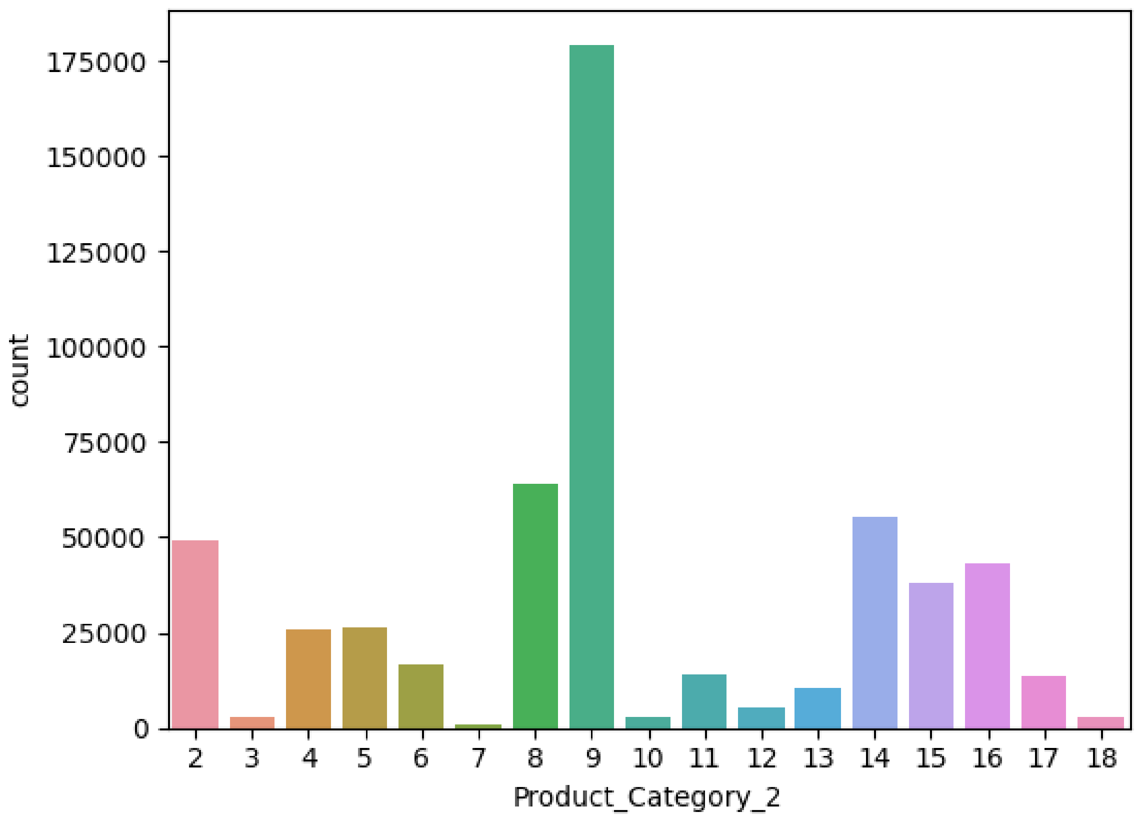
```
sns.kdeplot(df['Product_Category_2'])
```

```
<Axes: xlabel='Product_Category_2', ylabel='Density'>
```



```
sns.countplot(x='Product_Category_2',data=df)
```

```
<Axes: xlabel='Product_Category_2', ylabel='count'>
```



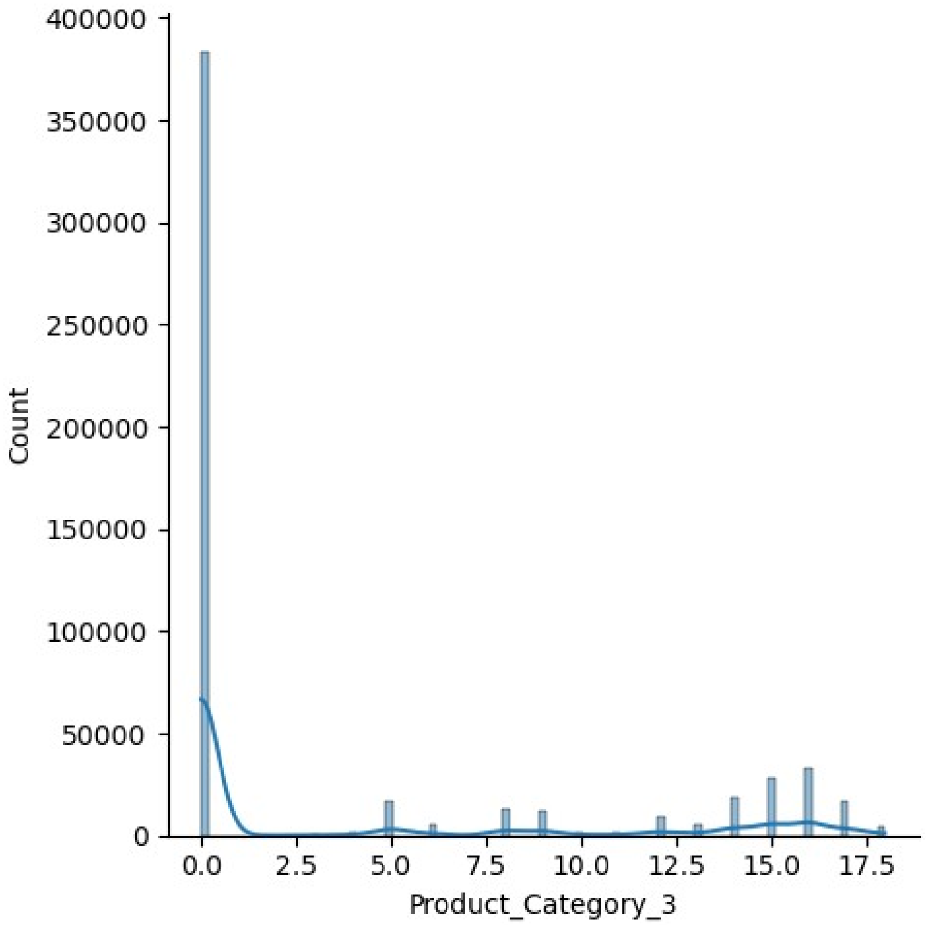
```
df.Product_Category_2.value_counts()

Product_Category_2
9     179331
8      64088
14     55108
2      49217
16     43255
15     37855
5      26235
4      25677
6      16466
11     14134
17     13320
13     10531
12      5528
10      3043
3       2884
18      2770
7        626
Name: count, dtype: int64

df.Product_Category_2.describe()
```

```
count    550068.000000
mean          9.576434
std           4.226025
min           2.000000
25%           8.000000
50%           9.000000
75%          14.000000
max          18.000000
Name: Product_Category_2, dtype: float64
```

```
sns.displot(df['Product_Category_3'],kde= True)
```

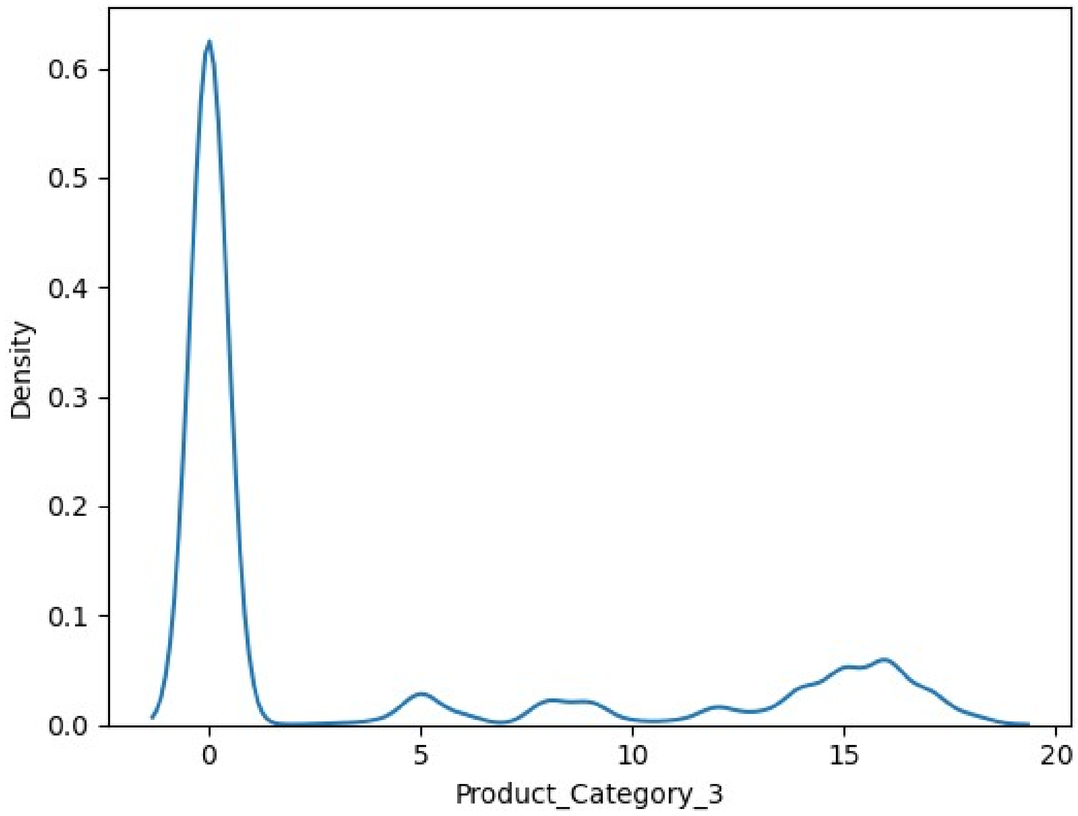
```
C:\Users\Lokesh\Python\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
<seaborn.axisgrid.FacetGrid at 0x23ea1d11f50>
```



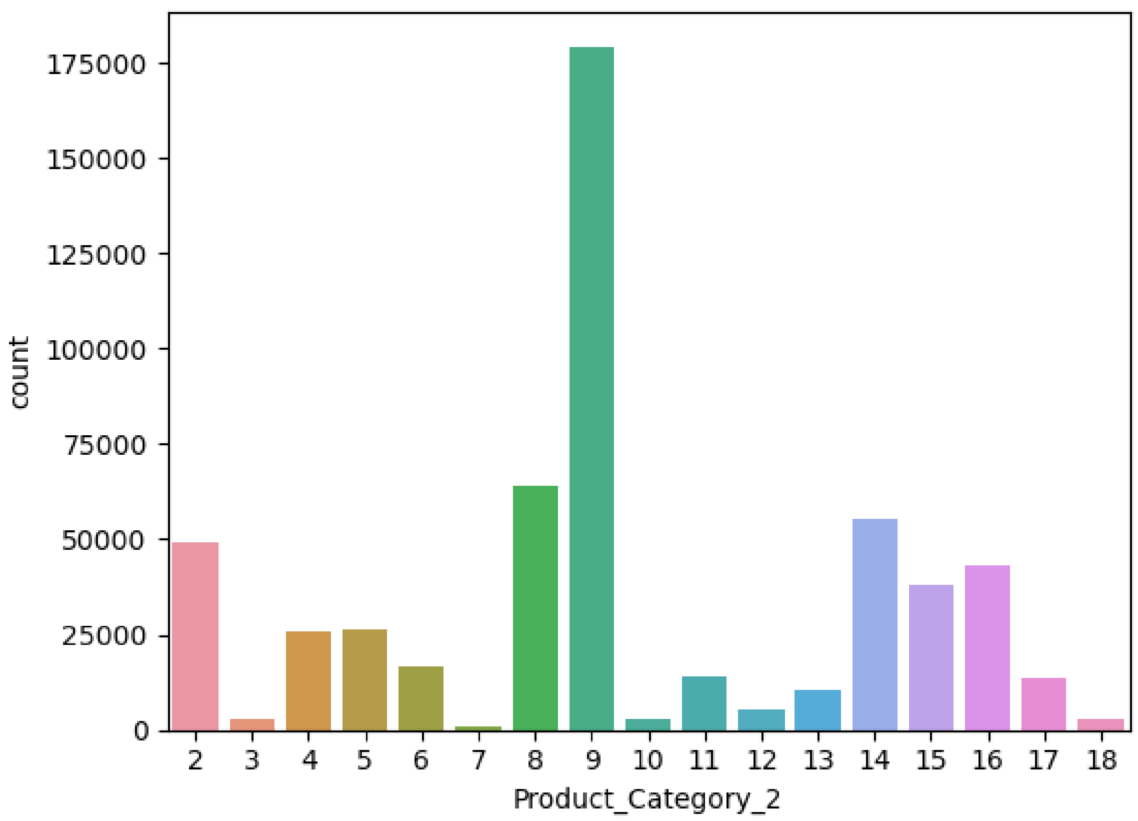
```
sns.kdeplot(df['Product_Category_3'])
```

```
<Axes: xlabel='Product_Category_3', ylabel='Density'>
```



```
sns.countplot(x='Product_Category_2',data=df)
```

```
<Axes: xlabel='Product_Category_2', ylabel='count'>
```



```
df.Product_Category_3.value_counts()
```

```
Product_Category_3
0     383247
16     32636
15     28013
14     18428
17     16702
5      16658
8      12562
9      11579
12      9246
13      5459
6       4890
18      4629
4       1875
11      1805
10      1726
3        613
Name: count, dtype: int64
```

```
df.Product_Category_3.describe()
```

```
count    550068.000000
mean          3.841941
```

```
std           6.250712
min           0.000000
25%           0.000000
50%           0.000000
75%           8.000000
max          18.000000
Name: Product_Category_3, dtype: float64
```

```
df["Gender"].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 550068 entries, 0 to 550067
Series name: Gender
Non-Null Count   Dtype
--------------   -----
550068 non-null  object
dtypes: object(1)
memory usage: 4.2+ MB
```

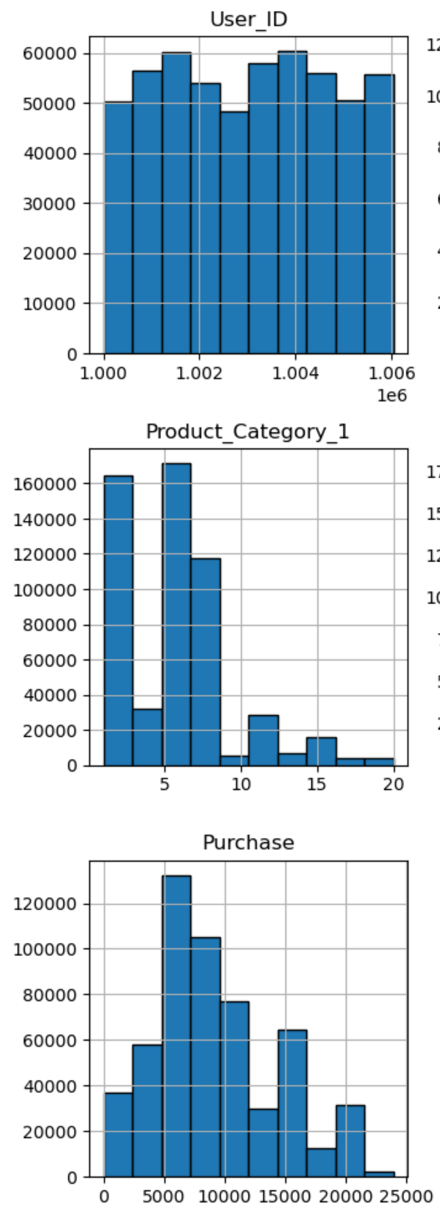
```
df['Gender'].value_counts()
```

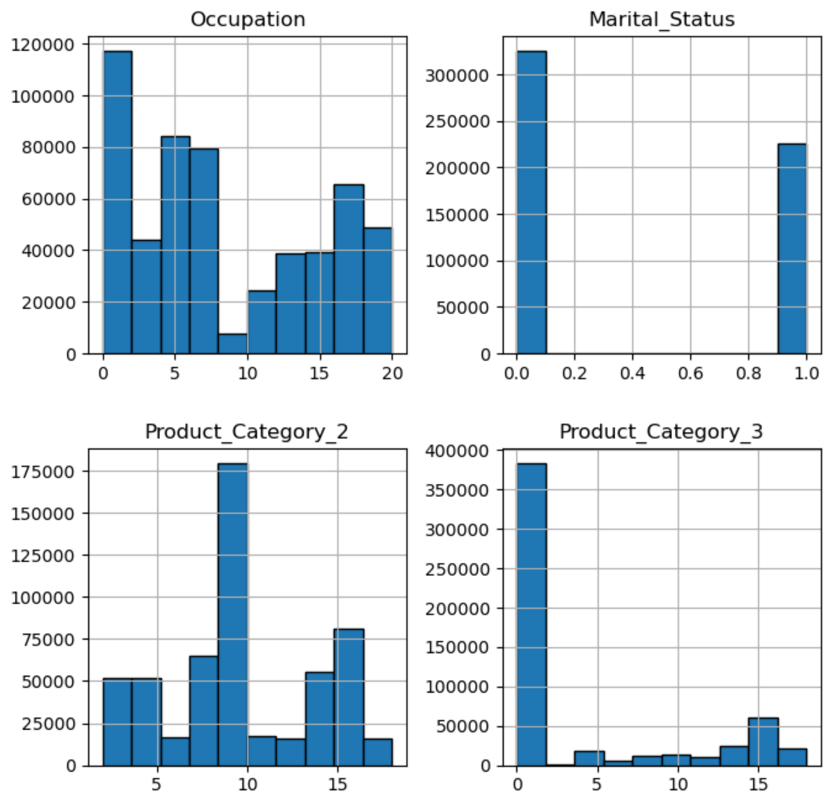
```
Gender
M    414259
F    135809
Name: count, dtype: int64
```

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

```
Occupation
4     72308
0     69638
7     59133
1     47426
17    40043
20    33562
12    31179
14    27309
2     26588
16    25371
6     20355
3     17650
10    12930
5     12177
15    12165
11    11586
19     8461
13     7728
18     6622
9      6291
8      1546
Name: count, dtype: int64
```

```
df.hist(edgecolor='black',figsize=(12,12));
```

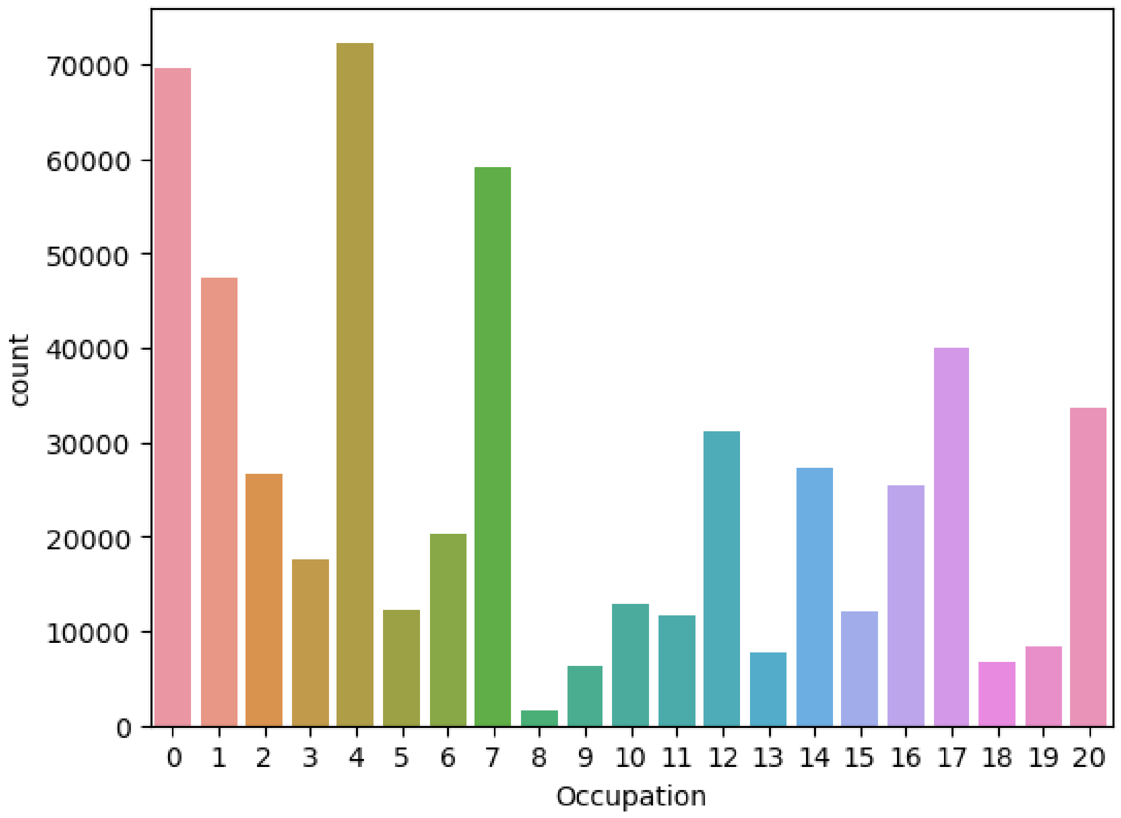




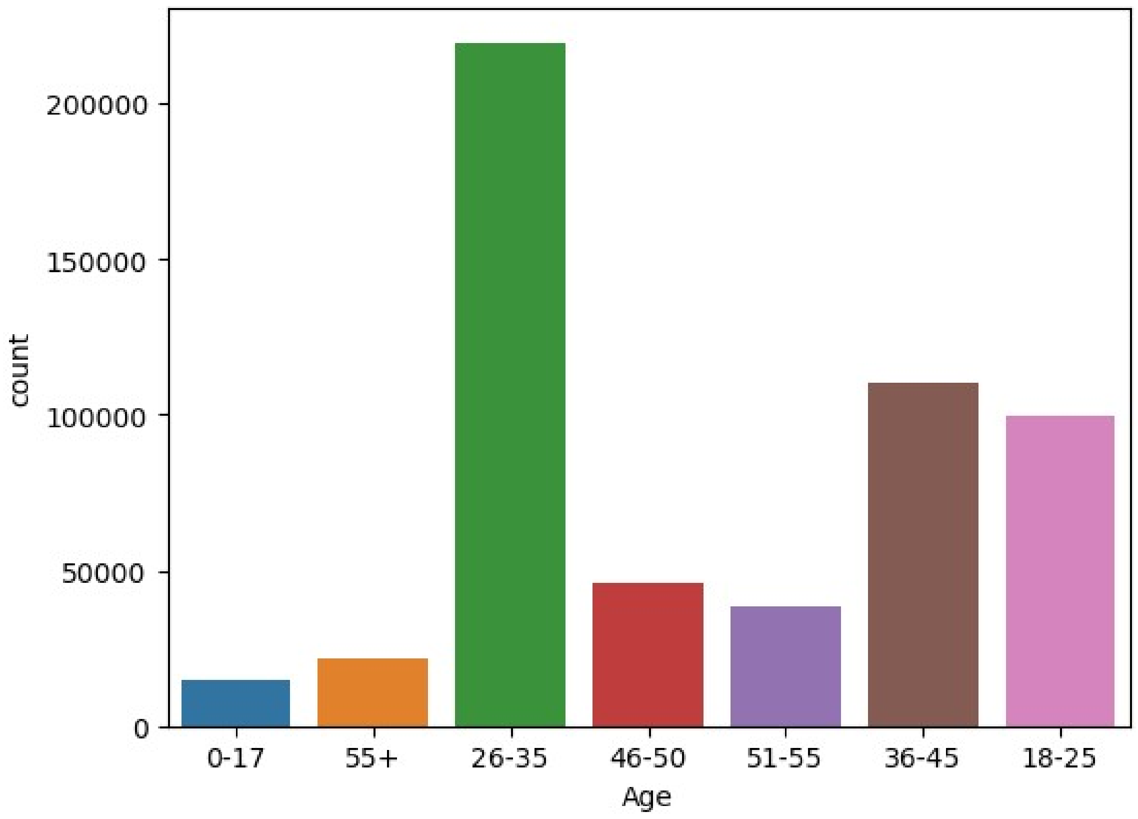
```
df.columns

Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation',
'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status',
'Product_Category_1',
       'Product_Category_2', 'Product_Category_3', 'Purchase'],
      dtype='object')
```

Occupation: In our dataset occupation 0,4 & 7 is high in count.

```
sns.countplot(x="Occupation", data=df)
plt.show()
```



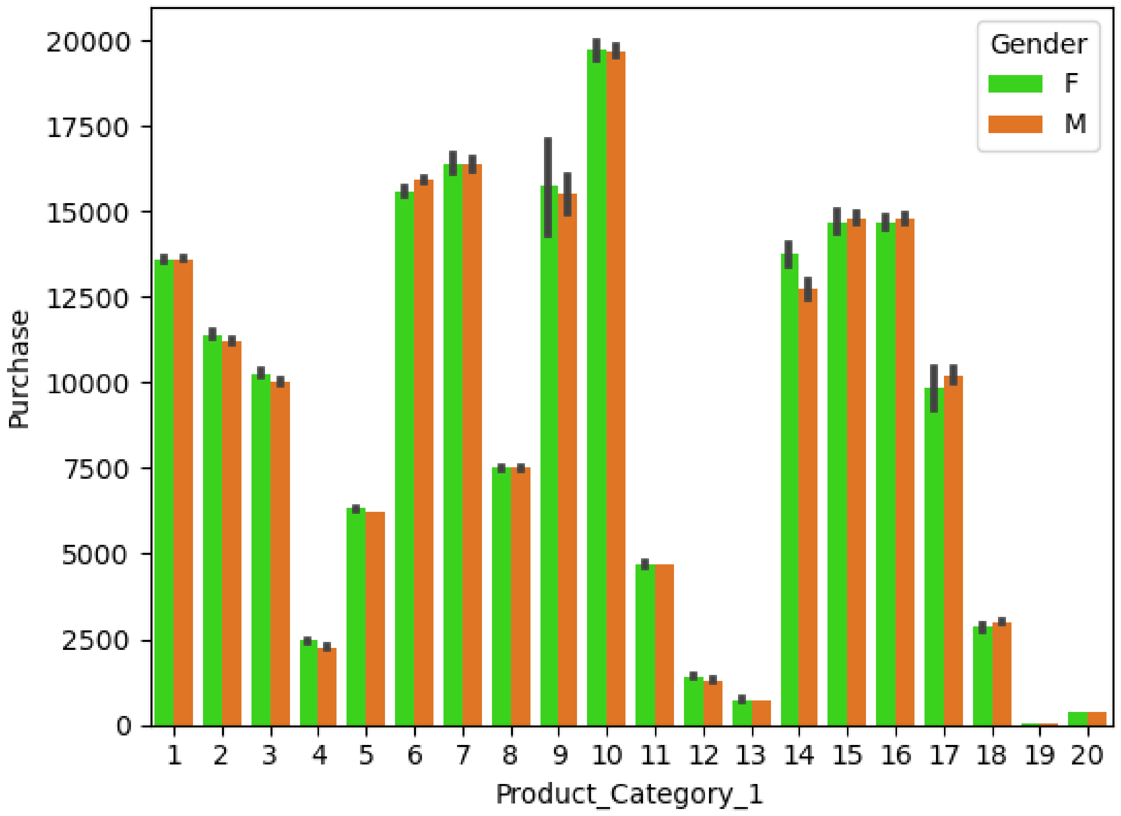
```
sns.countplot(x="Age", data=df)
plt.show()
```



How many purchases are made by people in Product_Category_1?

```
sns.barplot(x = "Product_Category_1",y  = "Purchase",hue =
"Gender",data = df,palette = "gist_ncar")
```

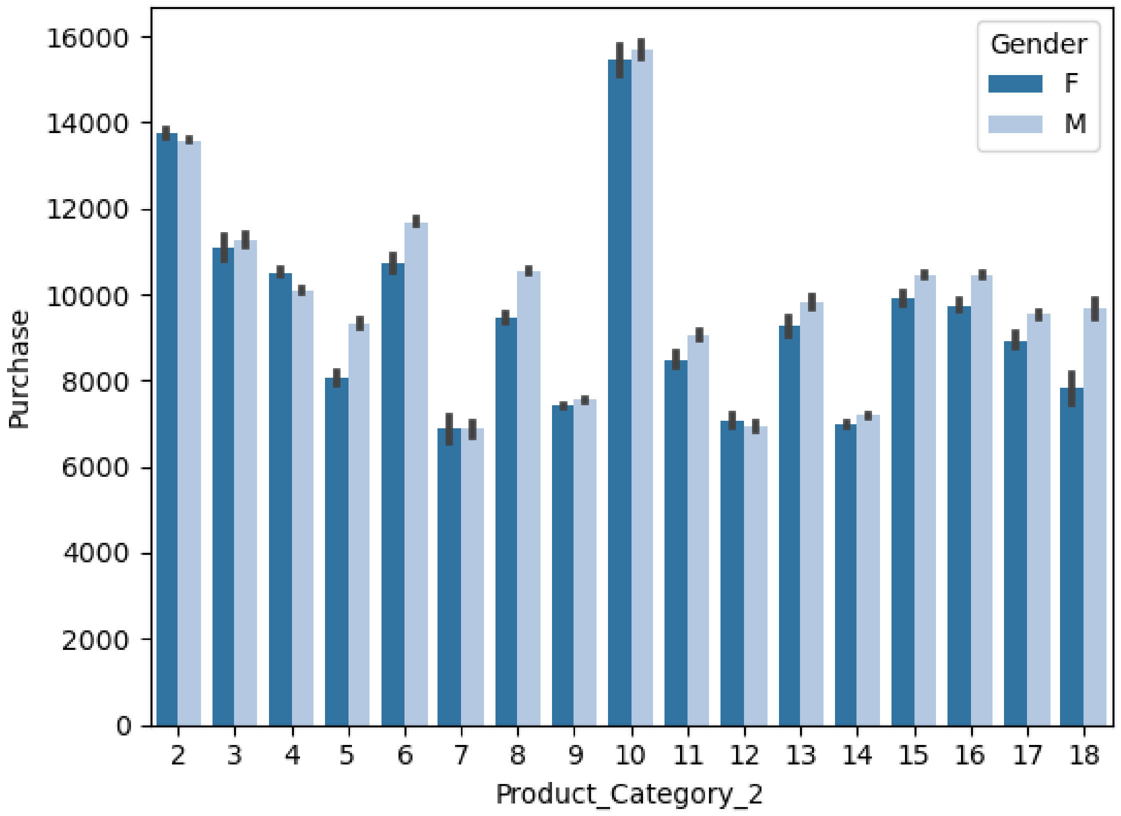
```
<Axes: xlabel='Product_Category_1', ylabel='Purchase'>
```



How many purchases are made by people in Product_Category_2?

```
sns.barplot(x = "Product_Category_2",y  = "Purchase",hue =
"Gender",data = df,palette = "tab20")
```

```
<Axes: xlabel='Product_Category_2', ylabel='Purchase'>
```



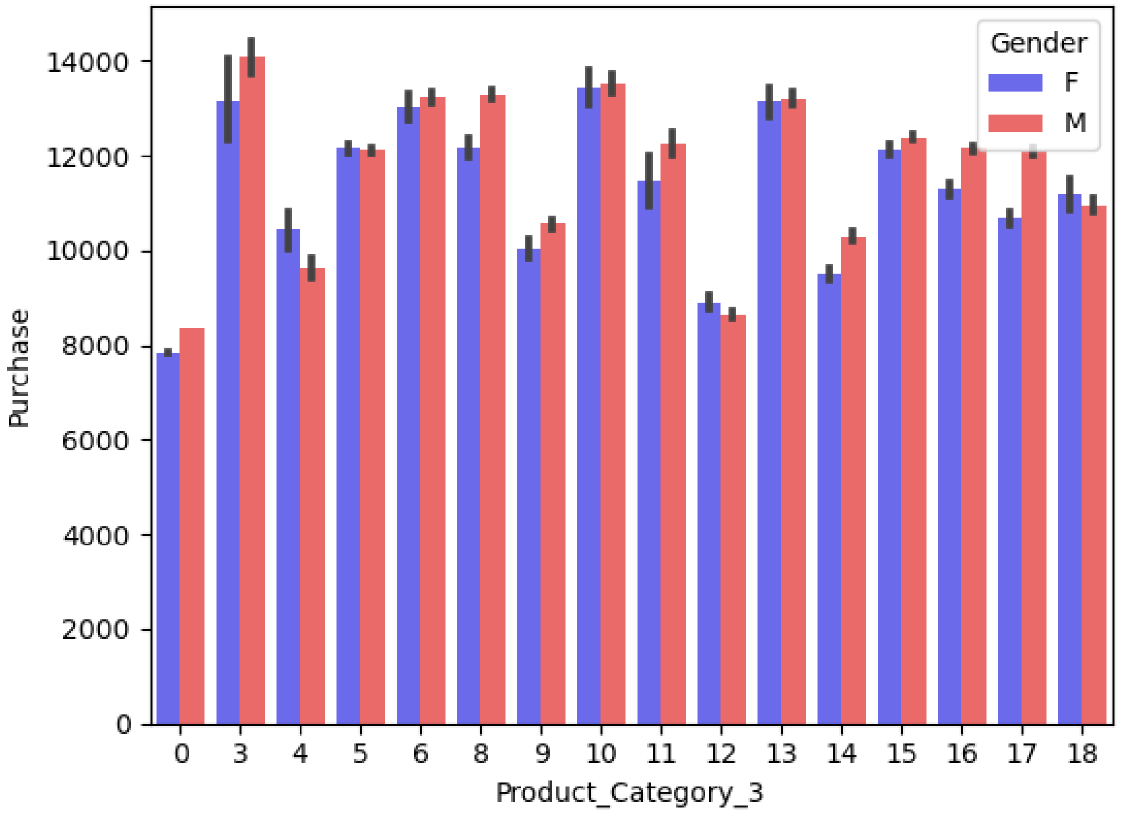
```
top_nreviews = df['Purchase'].nlargest(n=5).index
top_nreviews

Index([87440, 93016, 370891, 292083, 321782], dtype='int64')

top_rating_df = df.iloc[top_nreviews]
top_rating_df

        User_ID Product_ID Gender    Age  Occupation City_Category  \
87440   1001474  P00052842      M  26-35           4             A
93016   1002272  P00052842      M  26-35           0             C
370891  1003160  P00052842      M  26-35          17             C
292083  1003045  P00052842      M  46-50           1             B
321782  1001577  P00052842      M    55+           0             C

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
87440                            2               1                  10

93016                            1               0                  10

370891                           3               0                  10

292083                           2               1                  10
```

```
321782                          1               1                  10


        Product_Category_2  Product_Category_3  Purchase
87440                   15                   0     23961
93016                   15                   0     23961
370891                  15                   0     23961
292083                  15                   0     23960
321782                  15                   0     23960
```

How many purchases are made by people in Product_Category_3?

```
sns.barplot(x = "Product_Category_3",y  = "Purchase",hue =
"Gender",data = df,palette = "seismic")
```

```
<Axes: xlabel='Product_Category_3', ylabel='Purchase'>
```



```
df.groupby(['Product_Category_3'], as_index=False)['Purchase'].size()
```

```
    Product_Category_3    size
0                    0  383247
1                    3     613
2                    4    1875
3                    5   16658
4                    6    4890
5                    8   12562
6                    9   11579
7                   10    1726
8                   11    1805
9                   12    9246
10                  13    5459
11                  14   18428
12                  15   28013
13                  16   32636
14                  17   16702
15                  18    4629
```

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

```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

```
def ages(value):
    if '0-17' in value:
        value=value.replace('0-17','child')
        return str(value)
    elif '26-35'in value:
        value=value.replace('26-35','adult')
        return str(value)
    elif '18-25'in value:
        value=value.replace('18-25','teenage')
        return str(value)
    elif '36-45'in value:
        value=value.replace('36-45','adult')
        return str(value)
    elif '46-50'in value:
        value=value.replace('46-50','adult')
        return str(value)
    elif '51-55'in value:
        value=value.replace('51-55','old')
        return str(value)
    else:
        value=value.replace('55+','old')
        return str(value)
df['Age']=df['Age'].apply(ages)
```

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

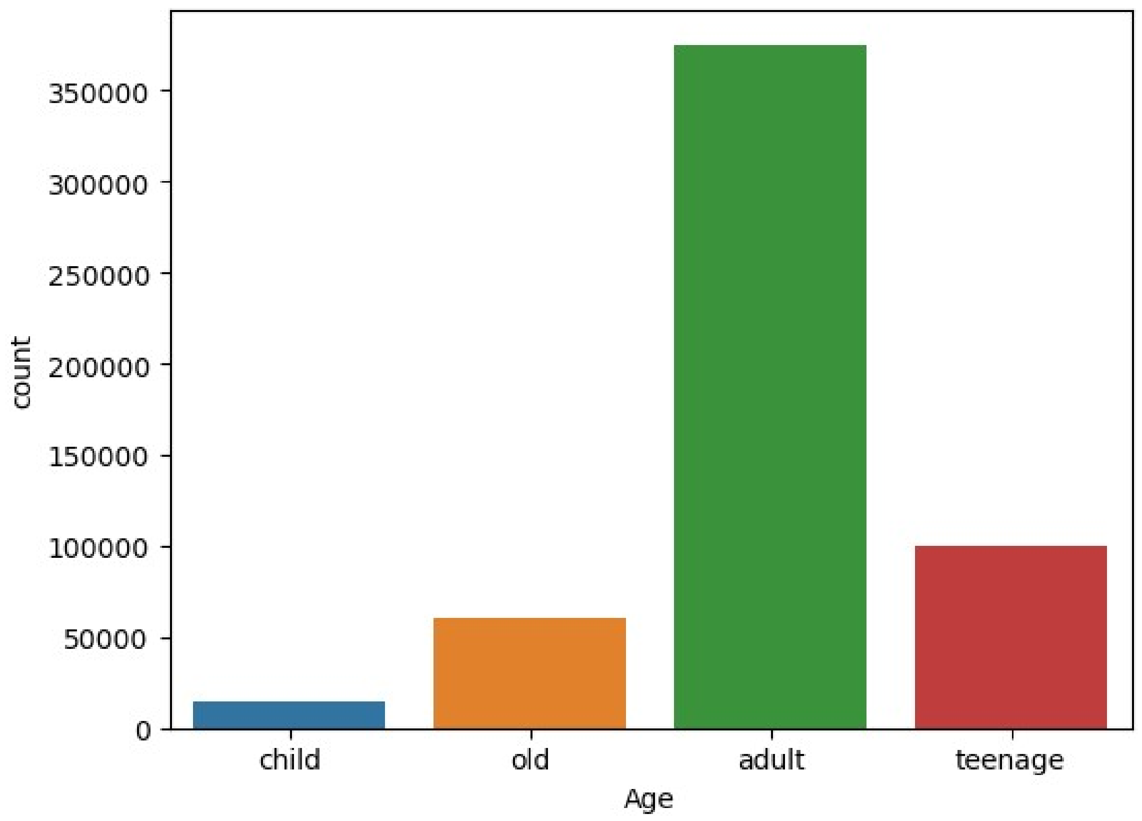
```
array(['child', 'old', 'adult', 'teenage'], dtype=object)
```

```
df.head(6)
```

```
   User_ID Product_ID Gender    Age  Occupation City_Category  \
0  1000001  P00069042      F  child          10             A
1  1000001  P00248942      F  child          10             A
2  1000001  P00087842      F  child          10             A
3  1000001  P00085442      F  child          10             A
4  1000002  P00285442      M    old          16             C
5  1000003  P00193542      M  adult          15             A

  Stay_In_Current_City_Years  Marital_Status  Product_Category_1  \
0                          2               0                   3
1                          2               0                   1
2                          2               0                  12
3                          2               0                  12
4                         4+               0                   8
5                          3               0                   1

   Product_Category_2  Product_Category_3  Purchase
0                   9                   0      8370
1                   6                  14     15200
2                   9                   0      1422
3                  14                   0      1057
4                   9                   0      7969
5                   2                   0     15227
```

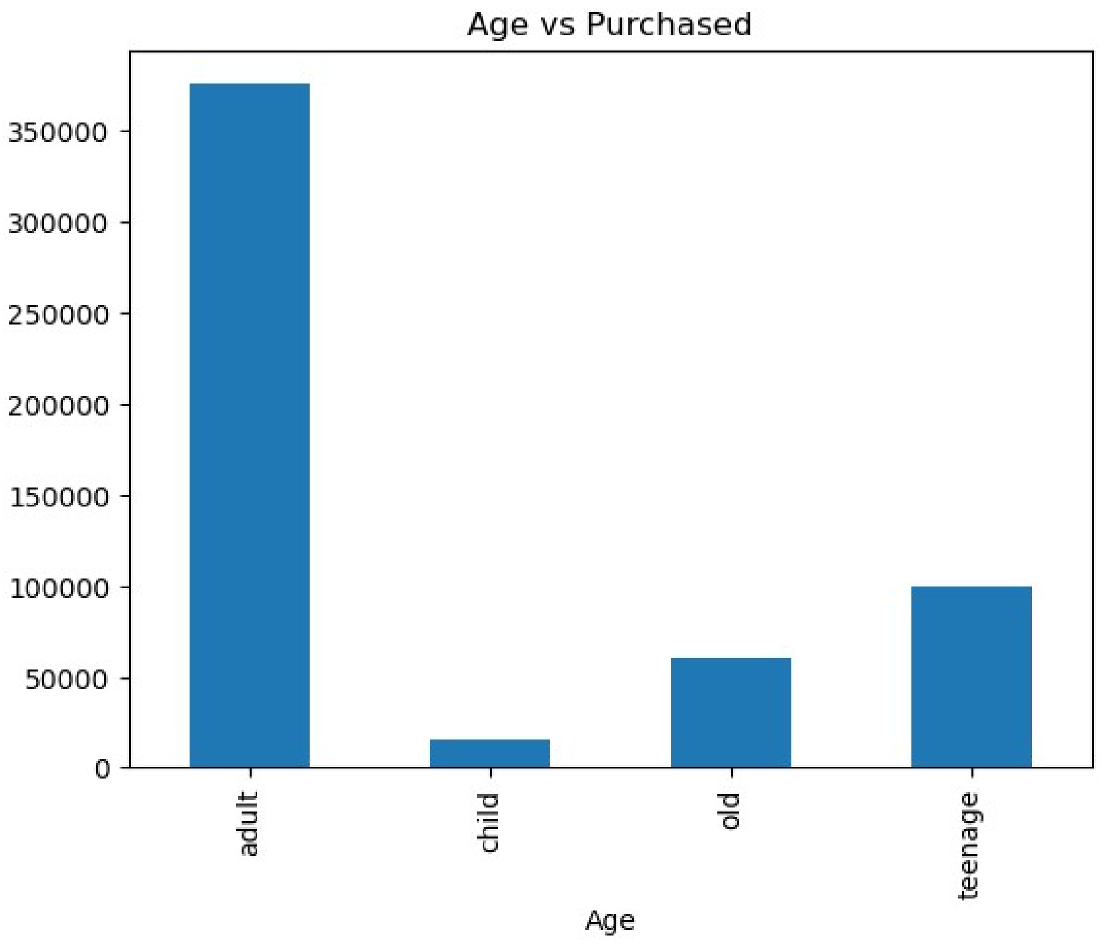
```
sns.countplot(x = "Age",data = df)
```

```
<Axes: xlabel='Age', ylabel='count'>
```



```
df.groupby(['Age'])['Purchase'].count().plot(kind='bar',title='Age vs
Purchased')
```

```
<Axes: title={'center': 'Age vs Purchased'}, xlabel='Age'>
```


Age vs Purchased
350000
300000
250000
200000
150000
100000
50000
0
adult
child
old
teenage
Age

Finish