Exploratory Data Analysis[EDA]

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
dataset=pd.read_csv("https://raw.githubusercontent.com//AnudipAE//DANLC//mas
dataset.head()
```

Out[7]:

	item_id	user_id	rating	timestamp	gender	category	brand	year	month
0	7	131	4	36692	Female	Home Audio	Philips	2000	6
1	19	231	5	36891	Female	Camera	Canon	2000	12
2	14	233	5	36893	Female	Camera	Kodak	2001	1
3	14	257	5	36926	Female	Camera	Kodak	2001	2
4	14	269	5	36952	Female	Camera	Kodak	2001	3

In [9]:
```
dataset.tail()
```

Out[9]:

	item_id	user_id	rating	timestamp	gender	category	brand	year
45161	7828	1157458	5	43341	Female	Headphones	Bose	2018
45162	8624	1157504	5	43342	Female	Headphones	Pyle	2018
45163	9513	1157527	5	43344	Male	Headphones	Mpow	2018
45164	9125	1157555	3	43348	Female	Headphones	EldHus	2018
45165	9478	1157632	1	43374	Female	Headphones	Etre Jeune	2018

In [11]:
```
dataset.shape
```

Out[11]: (45166, 12)

In [13]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45166 entries, 0 to 45165
Data columns (total 12 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   item_id    45166 non-null  int64 
 1   user_id    45166 non-null  int64 
 2   rating     45166 non-null  int64 
 3   timestamp  45166 non-null  int64 
 4   gender     45166 non-null  object
 5   category   45166 non-null  object
 6   brand      45166 non-null  object
 7   year       45166 non-null  int64 
 8   month      45166 non-null  int64 
 9   quantity   45166 non-null  int64 
 10  unitprice  45166 non-null  int64 
 11  amount     45166 non-null  int64 
dtypes: int64(9), object(3)
memory usage: 4.1+ MB
```

In [15]:
```
dataset['rating'].describe()
```

Out[15]:
```
count    45166.000000
mean         4.218594
std          1.221118
min          1.000000
25%          4.000000
50%          5.000000
75%          5.000000
max          5.000000
Name: rating, dtype: float64
```

In [17]:
```
dataset.nunique()
```

Out[17]:
```
item_id       1892
user_id      40401
rating           5
timestamp     4179
gender           2
category        10
brand           50
year            19
month           12
quantity         6
unitprice     5001
amount       19611
dtype: int64
```

In [30]:
```
# Dealing With Missing Values
```

In [19]:
```
dataset.isnull().sum()
```

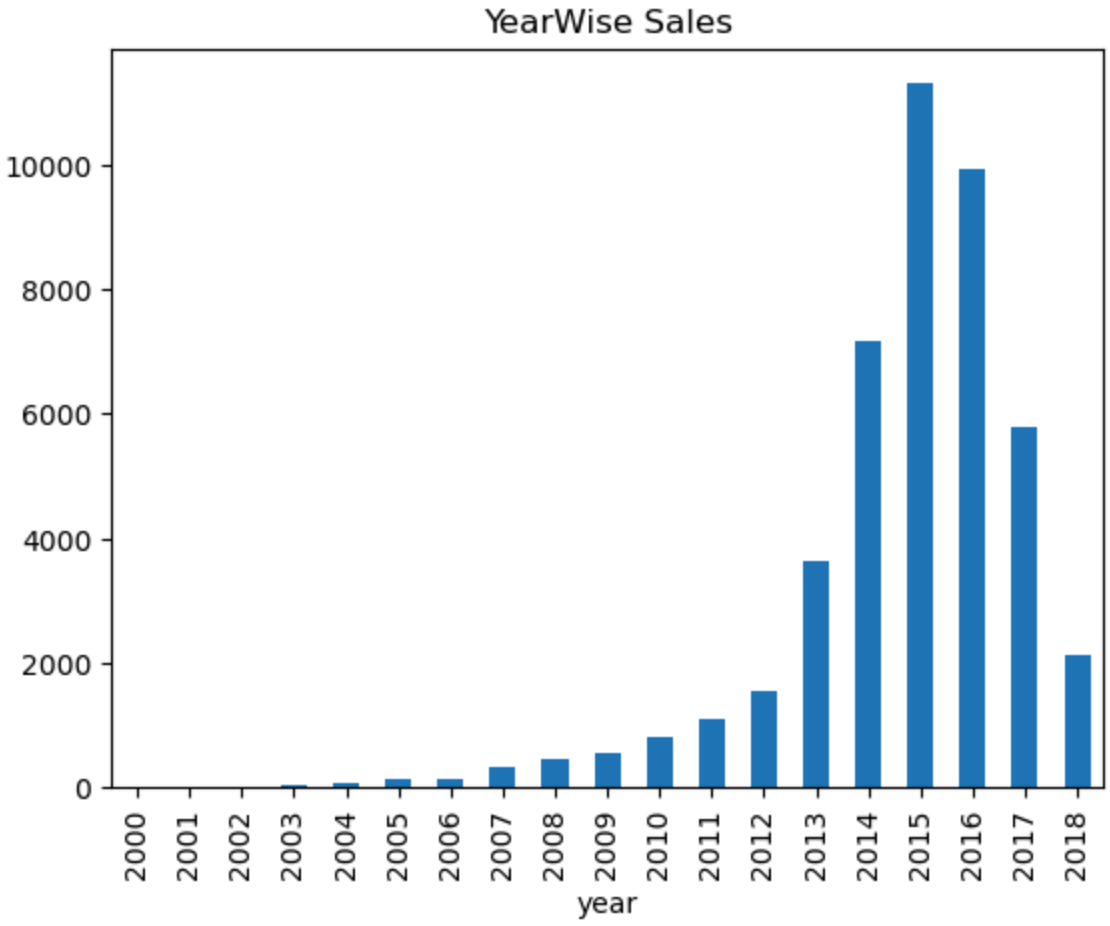
Out[19]:
```
item_id      0
user_id      0
rating       0
timestamp    0
gender       0
category     0
brand        0
year         0
month        0
quantity     0
unitprice    0
amount       0
dtype: int64
```

Finding Answers with the Data Using Visualizations

In [ ]:
```
# what was the best year of sales
```

In [40]:
```
dataset.groupby('year')['amount'].count().plot(kind='bar',title='YearWise Sa
```

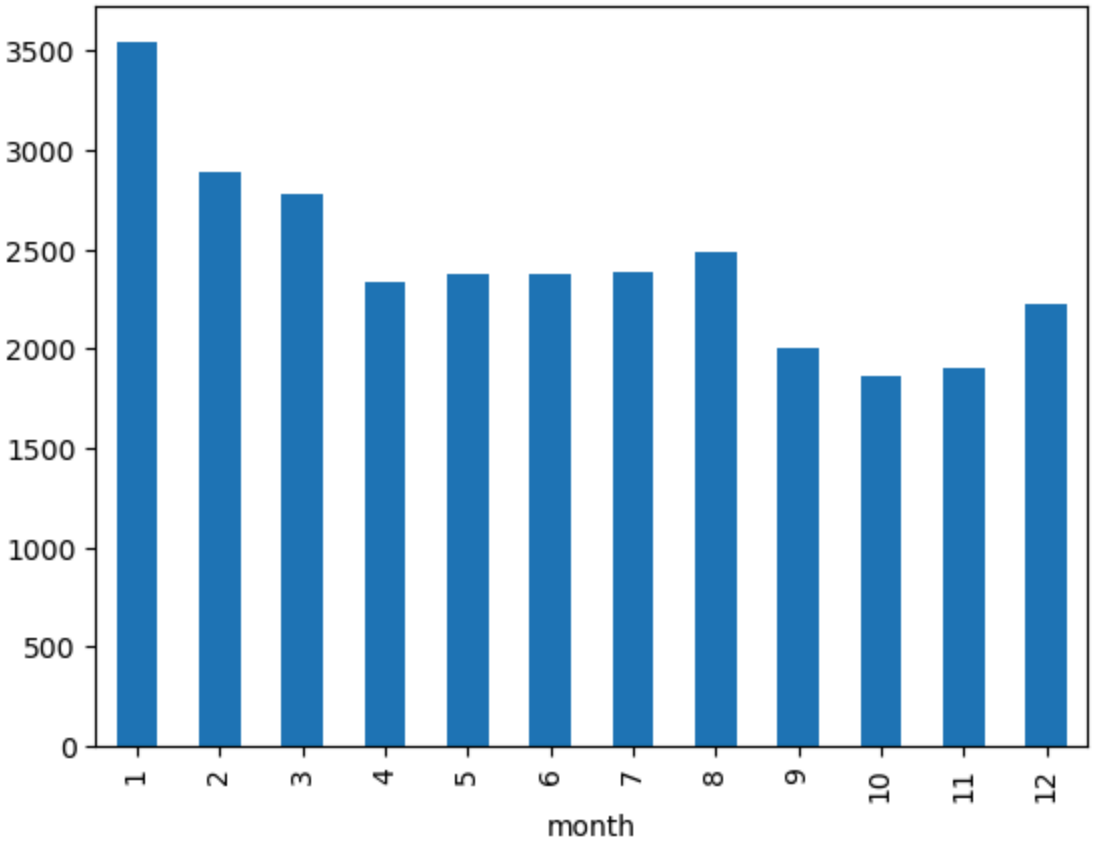
Out[40]:
```
<Axes: title={'center': 'YearWise Sales'}, xlabel='year'>
```



In [ ]:
```
# Which was the best month for sales between 2015 t0 2018
```

In [42]:
```
dataset_2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year']<= 2
dataset_2015_2018.groupby('month')['rating'].count().plot(kind='bar')
```

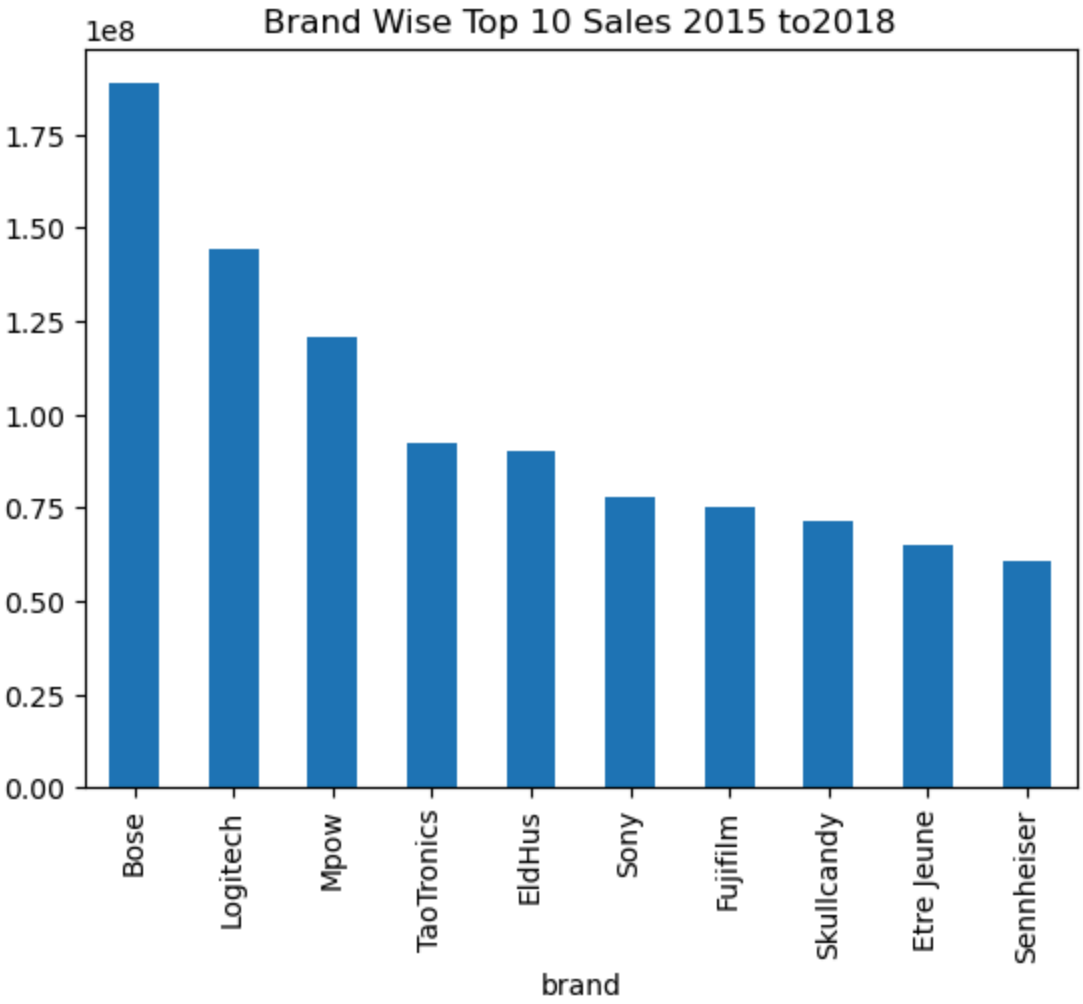
Out[42]:
```
<Axes: xlabel='month'>
```



In [ ]:
```
# what brand sold the most in 2015 to 2018
```

In [52]:
```
dataset_2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year'] <=
dataset_2015_2018.groupby('brand')['amount'].sum().sort_values(ascending=Fal
.plot(kind='bar',title='Brand Wise Top 10 Sales 2015 to2018',y='amount')
```

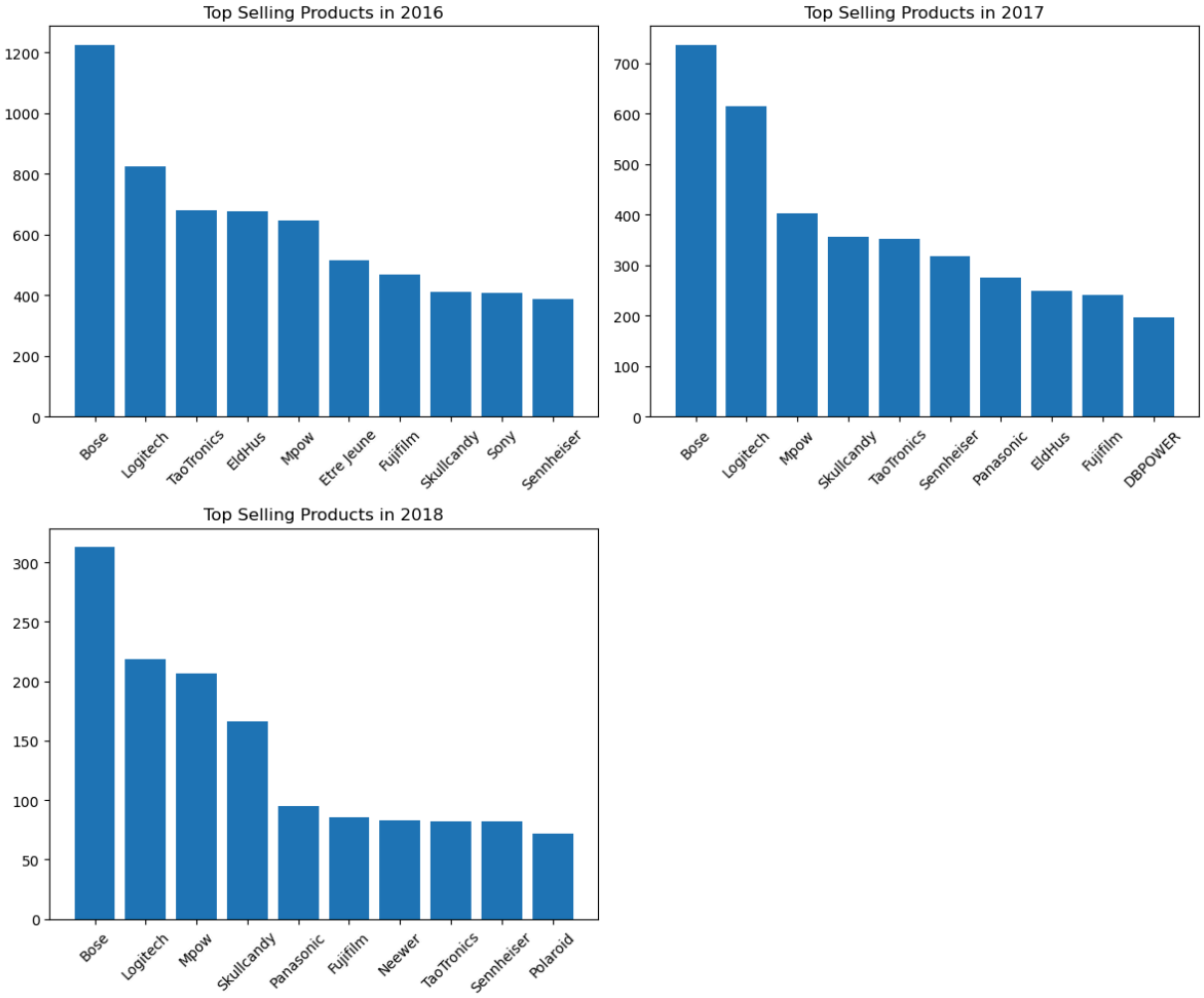
Out[52]:
```
<Axes: title={'center': 'Brand Wise Top 10 Sales 2015 to2018'}, xlabel='bra
nd'>
```



In [ ]:
```
# What products sold the most in the three years 2016, 2017 & 2018
```

In [64]:
```
 # Create subplots with 2 rows and 2 column
fig, axs = plt.subplots(2, 2, figsize=(12, 10))
# Plot for 2016
top_selling_2016 = dataset[dataset['year'] ==2016].groupby('brand')['rating'
axs[0, 0].bar(top_selling_2016.index, top_selling_2016)
axs[0, 0].set_title('Top Selling Products in 2016')
axs[0, 0].tick_params(axis='x', rotation=45) # Rotate x-axis labels
 # Plot for 2017
top_selling_2017 = dataset[dataset['year'] ==2017].groupby('brand')['rating'
axs[0, 1].bar(top_selling_2017.index, top_selling_2017)
axs[0, 1].set_title('Top Selling Products in 2017')
axs[0, 1].tick_params(axis='x', rotation=45) # Rotate x-axis labels
 # Plot for 2018
top_selling_2018 = dataset[dataset['year'] ==2018].groupby('brand')['rating'
axs[1, 0].bar(top_selling_2018.index, top_selling_2018)
axs[1, 0].set_title('Top Selling Products in 2018')
axs[1, 0].tick_params(axis='x', rotation=45) # Rotate x-axis labels
 # Hide the empty subplot
axs[1, 1].axis('off')
 # Adjust layout for better appearance
plt.tight_layout()
```

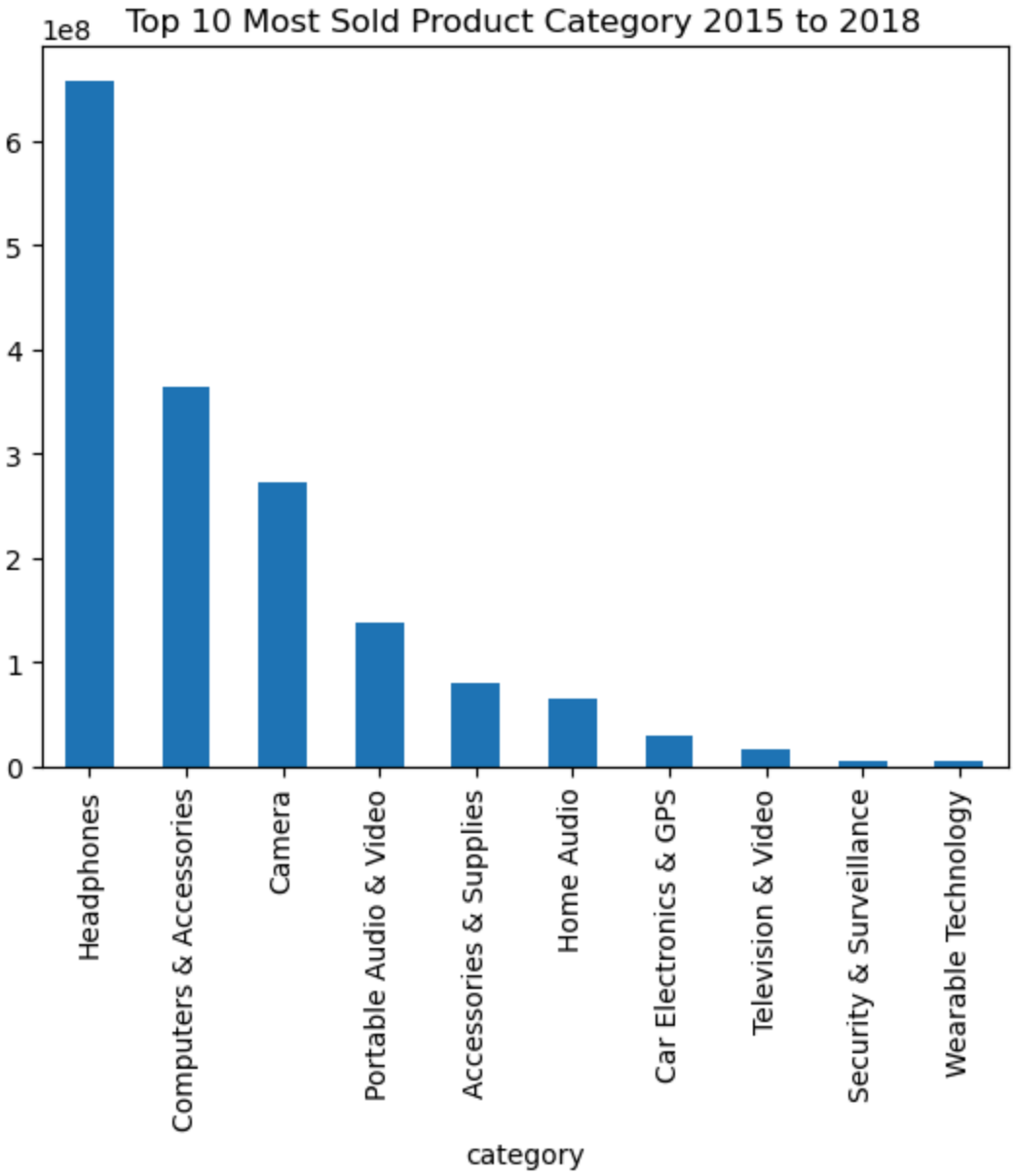
```
 # Show the plots
plt.show()
```



In [ ]:
```
# What product by category sold the most between 2015 to 2018?
```

In [74]:
```
dataset2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year']<= 20
dataset2015_2018.groupby('category')['amount'].sum().sort_values(ascending=F
```

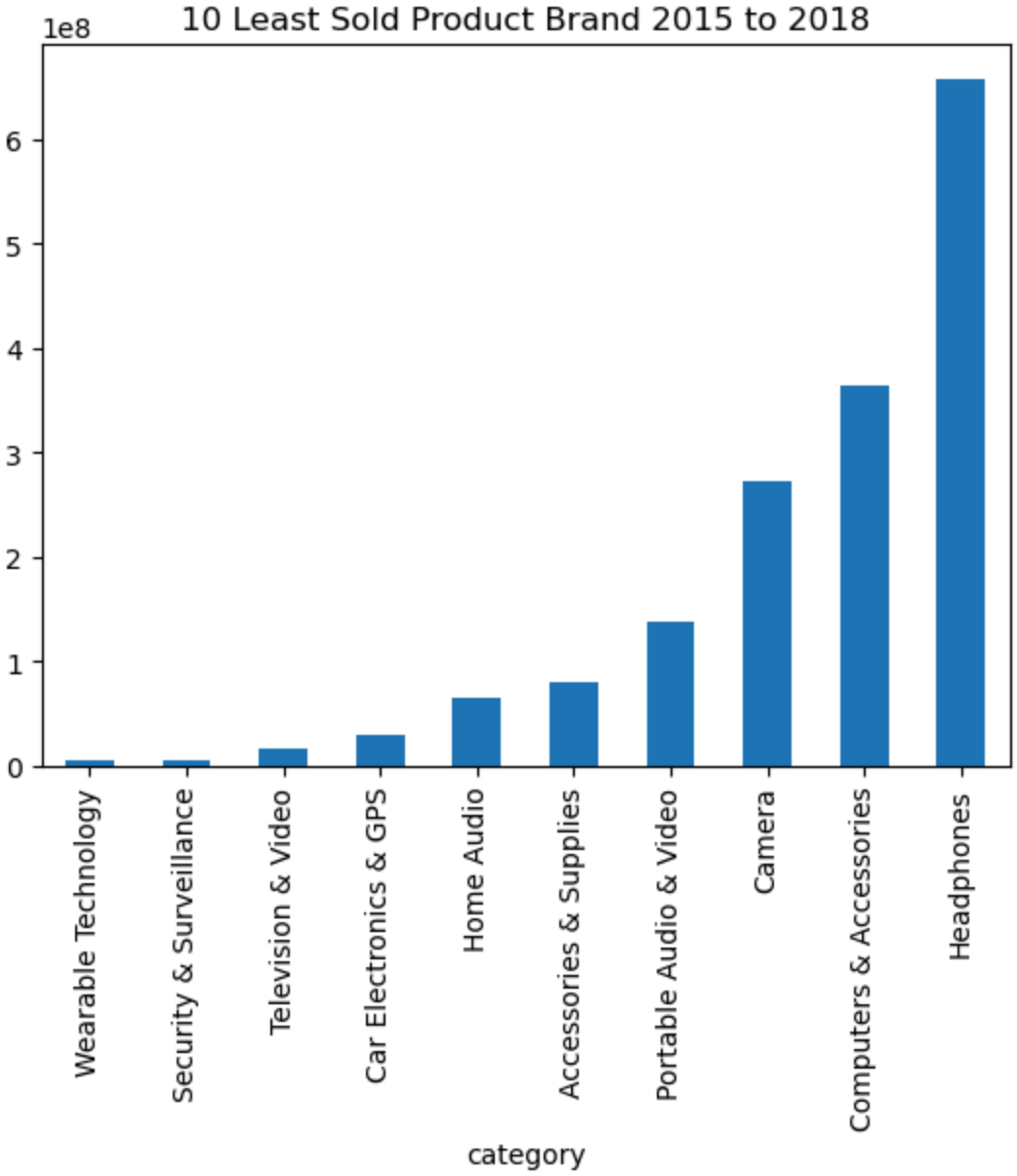
Out[74]:
```
<Axes: title={'center': 'Top 10 Most Sold Product Category 2015 to 2018'},
xlabel='category'>
```



In [ ]:
```
# What product by category sold the least between 2015 to 2018?
```

In [76]:
```
dataset2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year']<= 20
dataset2015_2018.groupby('category')['amount'].sum().sort_values(ascending=T
```

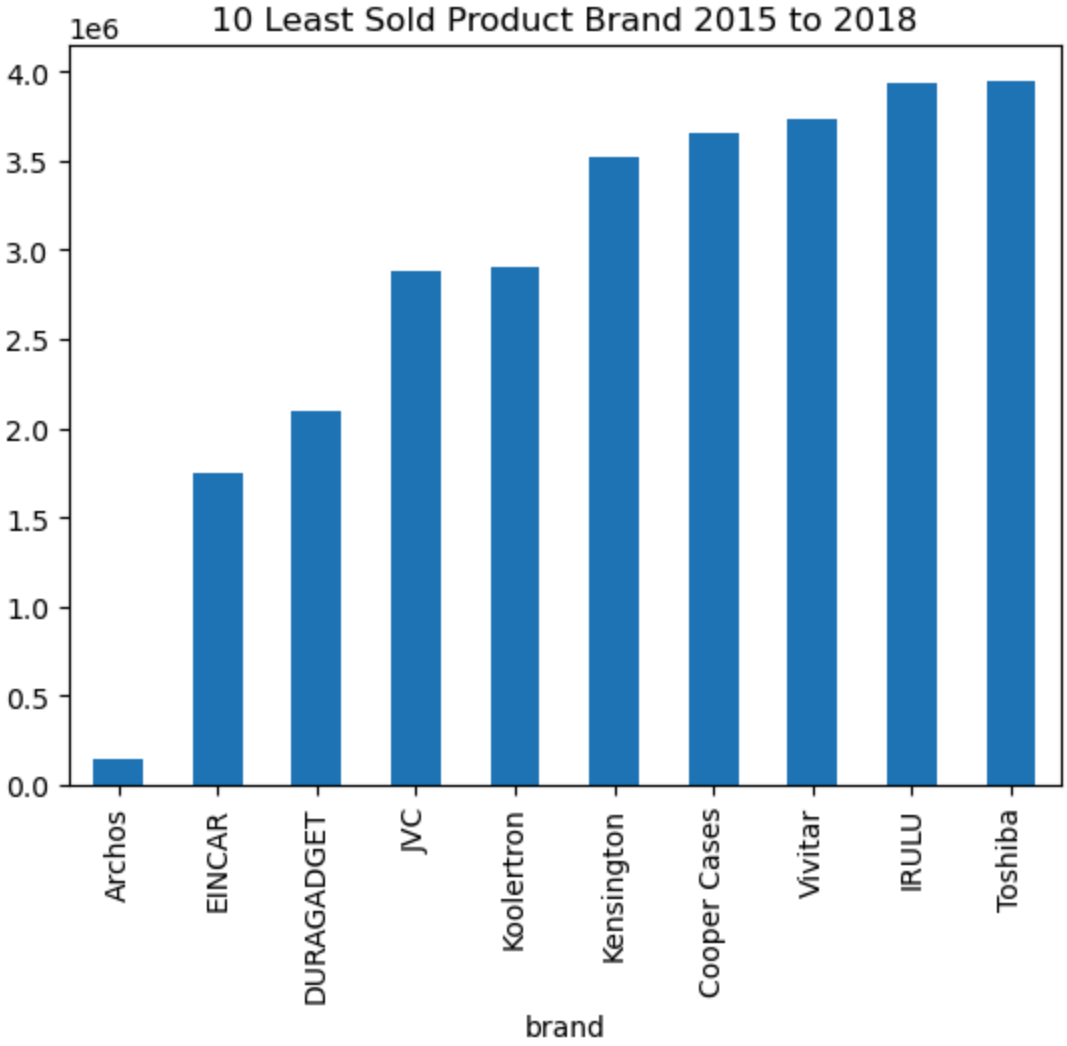
Out[76]:
```
<Axes: title={'center': '10 Least Sold Product Brand 2015 to 2018'}, xlabel
='category'>
```



In [ ]:
```
#  What product by brand name sold the least between 2015 to 2018?
```

In [78]:
```
dataset2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year']<= 20
dataset2015_2018.groupby('brand')['amount'].sum().sort_values(ascending=True
```

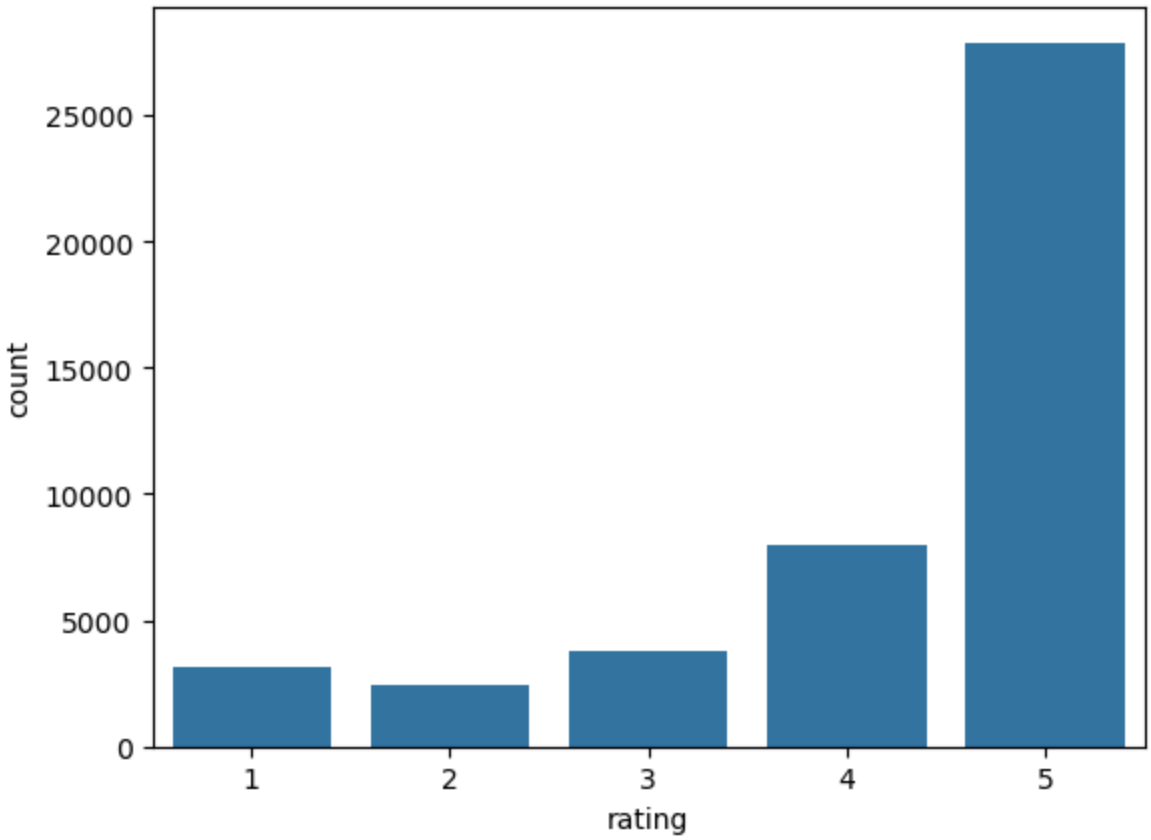
Out[78]:
```
<Axes: title={'center': '10 Least Sold Product Brand 2015 to 2018'}, xlabel
='brand'>
```



In [ ]:
```
# Ratings Distribution
```

In [80]:
```
sns.countplot(x='rating', data=dataset)
```

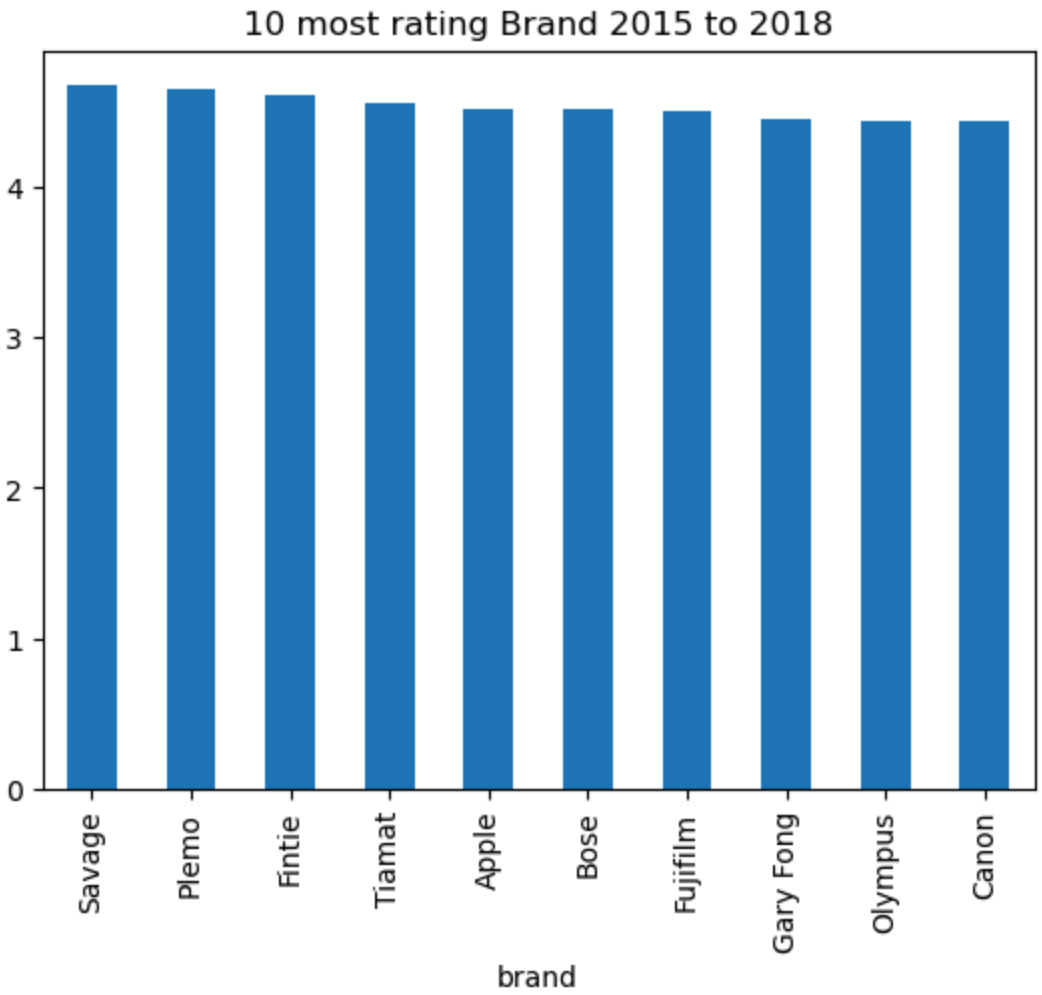
Out[80]:
```
<Axes: xlabel='rating', ylabel='count'>
```



In [ ]:
```
# Best rated brands
```

In [82]:
```
dataset2015_2018 = dataset[(dataset['year'] >= 2015) & (dataset['year']<= 20
dataset2015_2018.groupby('brand')['rating'].mean().sort_values(ascending=Fal
```

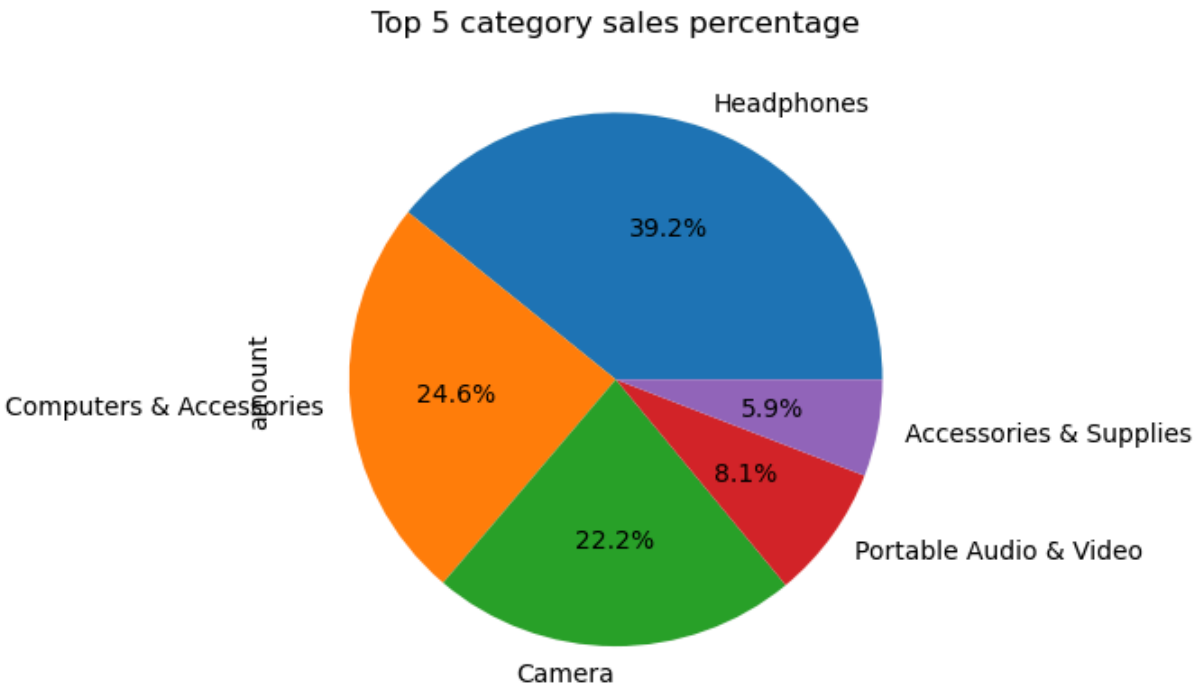
Out[82]:
```
<Axes: title={'center': '10 most rating Brand 2015 to 2018'}, xlabel='bran
d'>
```



In [ ]:
```
# Top 5 category sales percentage
```

In [84]:
```
dataset.groupby('category')['amount'].sum().sort_values(ascending=False).hea
```

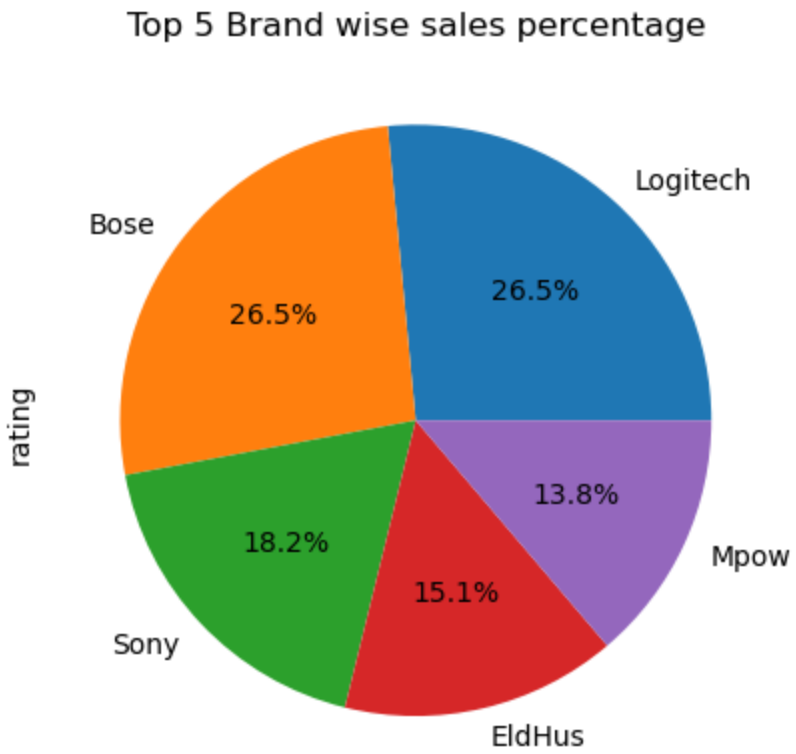
Out[84]:
```
<Axes: title={'center': 'Top 5 category sales percentage'}, ylabel='amoun
t'>
```



In [ ]:
```
# Brand wise sales percentage
```

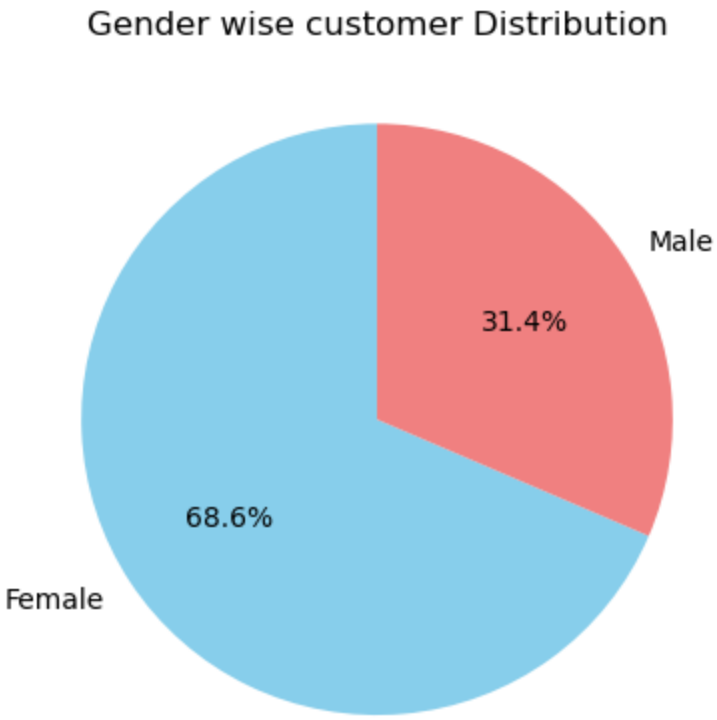
In [88]:
```
dataset.groupby('brand')['rating'].count().sort_values(ascending=False).heac
```

Out[88]:
```
<Axes: title={'center': 'Top 5 Brand wise sales percentage'}, ylabel='ratin
g'>
```



In [ ]:
```
# Gender wise customer distribution
```

In [90]:
```
gender_distribution = dataset['gender'].value_counts()
plt.pie(gender_distribution, labels=gender_distribution.index,autopct='%1.1f
plt.title('Gender wise customer Distribution')
plt.show()
```



In [ ]: