In [1]:
```
import pandas as pd
import numpy as np
df=pd.read_csv("../Documents/sales_data_sample.csv",encoding= 'unicode_escape')
```

In [2]:
```
df.head(5)
```

Out[2]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDA
0	10107	30	95.70	2	2871.00	2/24/20 0:
1	10121	34	81.35	5	2765.90	5/7/20 0:
2	10134	41	94.74	2	3884.34	7/1/20 0:
3	10145	45	83.26	6	3746.70	8/25/20 0:
4	10159	49	100.00	14	5205.27	10/10/20 0:

5 rows × 25 columns

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

In [4]:
```
raw_data = df.dropna(axis=0)
```

In [5]:
```
df.isnull().sum()
```

Out[5]:

```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

In [19]:

```
X = df[['SALES','PRODUCTCODE']]
X['SALES'] = X['SALES'].astype(int)
```

```
C:\Users\siddh\AppData\Local\Temp\ipykernel_10608\441610815.py:2: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['SALES'] = X['SALES'].astype(int)
```

Out[19]:

	SALES	PRODUCTCODE
0	2871	S10_1678
1	2765	S10_1678
2	3884	S10_1678
3	3746	S10_1678
4	5205	S10_1678
...	...	...
2818	2244	S72_3212
2819	3978	S72_3212
2820	5417	S72_3212
2821	2116	S72_3212
2822	3079	S72_3212

2823 rows × 2 columns

In [22]:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
X['PRODUCTCODE'] = le.fit_transform (X['PRODUCTCODE'])
X
```

```
C:\Users\siddh\AppData\Local\Temp\ipykernel_10608\3014778578.py:4: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['PRODUCTCODE'] = le.fit_transform (X['PRODUCTCODE'])
```

Out[22]:

	SALES	PRODUCTCODE
0	2871	0
1	2765	0
2	3884	0
3	3746	0
4	5205	0
...	...	...
2818	2244	108
2819	3978	108
2820	5417	108
2821	2116	108
2822	3079	108

2823 rows × 2 columns

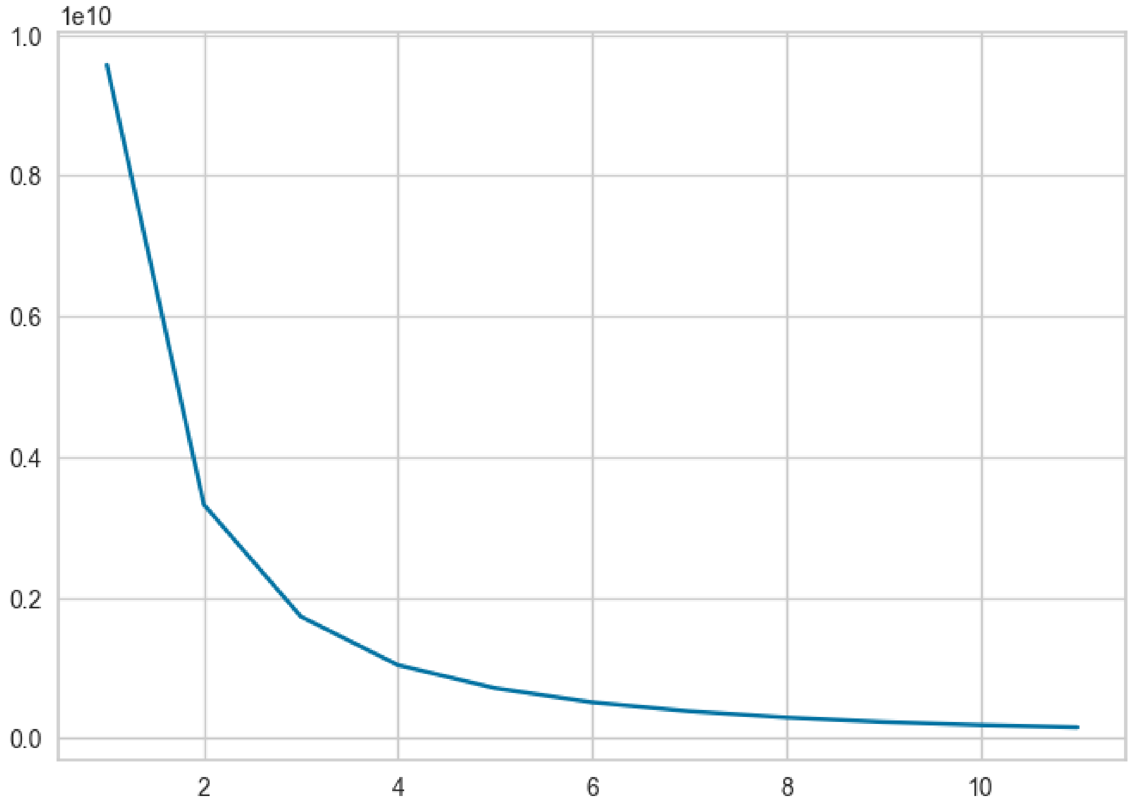
In [28]:
```
from sklearn.cluster import KMeans
wcss=[]

for i in range(1,12):
    clustering = KMeans(n_clusters=i)
    clustering.fit(X)
    wcss.append(clustering.inertia_)

ks=[1,2,3,4,5,6,7,8,9,10,11]
```

In [29]:
```
import seaborn as sb
sb.lineplot(x=ks , y=wcss)
```

Out[29]:
```
<AxesSubplot: >
```



In [30]:
```
kmeans=KMeans(4).fit(X)
labels=kmeans.labels_
```

In [32]:
```
from collections import Counter
Counter(kmeans.labels_)
```

Out[32]:
```
Counter({3: 1024, 0: 565, 1: 1035, 2: 199})
```

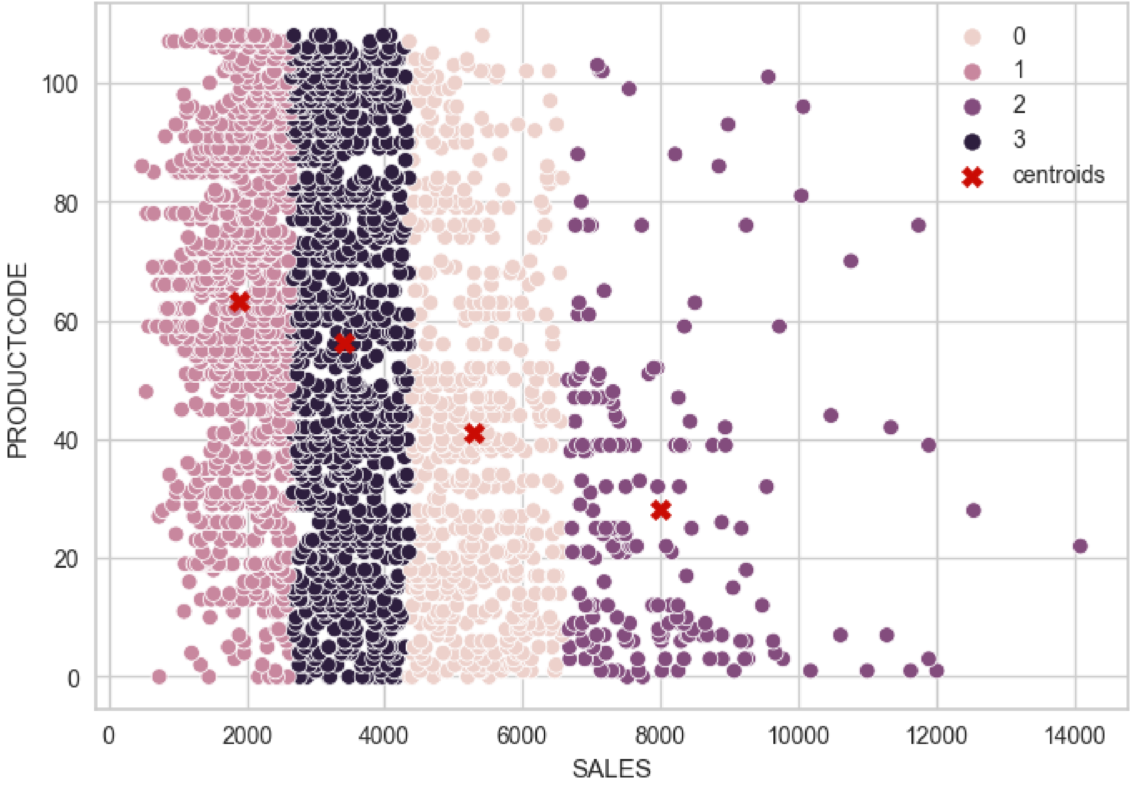
In [33]:
```
kmeans.cluster_centers_
```

Out[33]:
```
array([[5289.27065026,   41.01230228],
       [1880.02224371,   63.28626692],
       [7983.1758794 ,   28.05025126],
       [3417.35455436,   56.26444662]])
```

In [34]:
```
import matplotlib.pyplot as plt

sb.scatterplot(data=df, x="SALES", y="PRODUCTCODE", hue=kmeans.labels_)
```

```
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            marker="X", c="r", s=80, label="centroids")
plt.legend()
plt.show()
```



```
In [ ]:
```