In [3]:
```
conda install -c districtdatalabs yellowbrick
```

```
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: C:\Users\SHUBHAM\anaconda3

  added / updated specs:
    - yellowbrick


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    yellowbrick-1.5            |  py311haa95532_0         353 KB
    ------------------------------------------------------------
                                           Total:         353 KB

The following NEW packages will be INSTALLED:

  yellowbrick        pkgs/main/win-64::yellowbrick-1.5-py311haa95532_0



Downloading and Extracting Packages

yellowbrick-1.5      | 353 KB    |            |   0%
yellowbrick-1.5      | 353 KB    | 4          |   5%
yellowbrick-1.5      | 353 KB    | #8         |  18%
yellowbrick-1.5      | 353 KB    | ###1       |  32%
yellowbrick-1.5      | 353 KB    | ####5      |  45%
yellowbrick-1.5      | 353 KB    | #####4     |  54%
yellowbrick-1.5      | 353 KB    | ######3    |  63%
yellowbrick-1.5      | 353 KB    | #######6   |  77%
yellowbrick-1.5      | 353 KB    | #########  |  91%
yellowbrick-1.5      | 353 KB    | ########## | 100%
yellowbrick-1.5      | 353 KB    | ########## | 100%



Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done

Note: you may need to restart the kernel to use updated packages.


==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.9.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.9.0
```

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import seaborn as sns
from sklearn import preprocessing
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans

from collections import Counter
```

In [5]:
```
data = pd.read_csv("sales_data_sample.csv", encoding='Latin-1')
print(data)
```

```
      ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0           10107               30      95.70                2  2871.00
1           10121               34      81.35                5  2765.90
2           10134               41      94.74                2  3884.34
3           10145               45      83.26                6  3746.70
4           10159               49     100.00               14  5205.27
...           ...              ...        ...              ...      ...
2818        10350               20     100.00               15  2244.40
2819        10373               29     100.00                1  3978.51
2820        10386               43     100.00                4  5417.57
2821        10397               34      62.24                1  2116.16
2822        10414               47      65.52                9  3079.44

            ORDERDATE    STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0      2/24/2003 0:00   Shipped       1         2     2003  ...
1       5/7/2003 0:00   Shipped       2         5     2003  ...
2       7/1/2003 0:00   Shipped       3         7     2003  ...
3      8/25/2003 0:00   Shipped       3         8     2003  ...
4     10/10/2003 0:00   Shipped       4        10     2003  ...
...               ...       ...     ...       ...      ...  ...
2818   12/2/2004 0:00   Shipped       4        12     2004  ...
2819   1/31/2005 0:00   Shipped       1         1     2005  ...
2820    3/1/2005 0:00  Resolved       1         3     2005  ...
2821   3/28/2005 0:00   Shipped       1         3     2005  ...
2822    5/6/2005 0:00   On Hold       2         5     2005  ...

                      ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0            897 Long Airport Avenue           NaN            NYC    NY
1                59 rue de l'Abbaye           NaN          Reims   NaN
2     27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3                78934 Hillside Dr.           NaN       Pasadena    CA
4                   7734 Strong St.           NaN  San Francisco    CA
...                             ...           ...            ...   ...
2818             C/ Moralzarzal, 86           NaN         Madrid   NaN
2819                    Torikatu 38           NaN           Oulu   NaN
2820             C/ Moralzarzal, 86           NaN         Madrid   NaN
2821          1 rue Alsace-Lorraine           NaN       Toulouse   NaN
2822             8616 Spinnaker Dr.           NaN         Boston    MA

     POSTALCODE  COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0         10022      USA       NaN             Yu             Kwai    Small
1         51100   France      EMEA        Henriot             Paul    Small
2         75508   France      EMEA       Da Cunha           Daniel   Medium
3         90003      USA       NaN          Young            Julie   Medium
4           NaN      USA       NaN          Brown            Julie   Medium
...         ...      ...       ...            ...              ...      ...
2818      28034    Spain      EMEA         Freyre            Diego    Small
2819      90110  Finland      EMEA      Koskitalo           Pirkko   Medium
2820      28034    Spain      EMEA         Freyre            Diego   Medium
2821      31000   France      EMEA         Roulet          Annette    Small
2822      51003      USA       NaN        Yoshido             Juri   Medium

[2823 rows x 25 columns]
```

In [6]:
```
print(data.shape)
```

```
(2823, 25)
```

In [7]:
```
print(data.isnull().sum() )
```

```
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
```

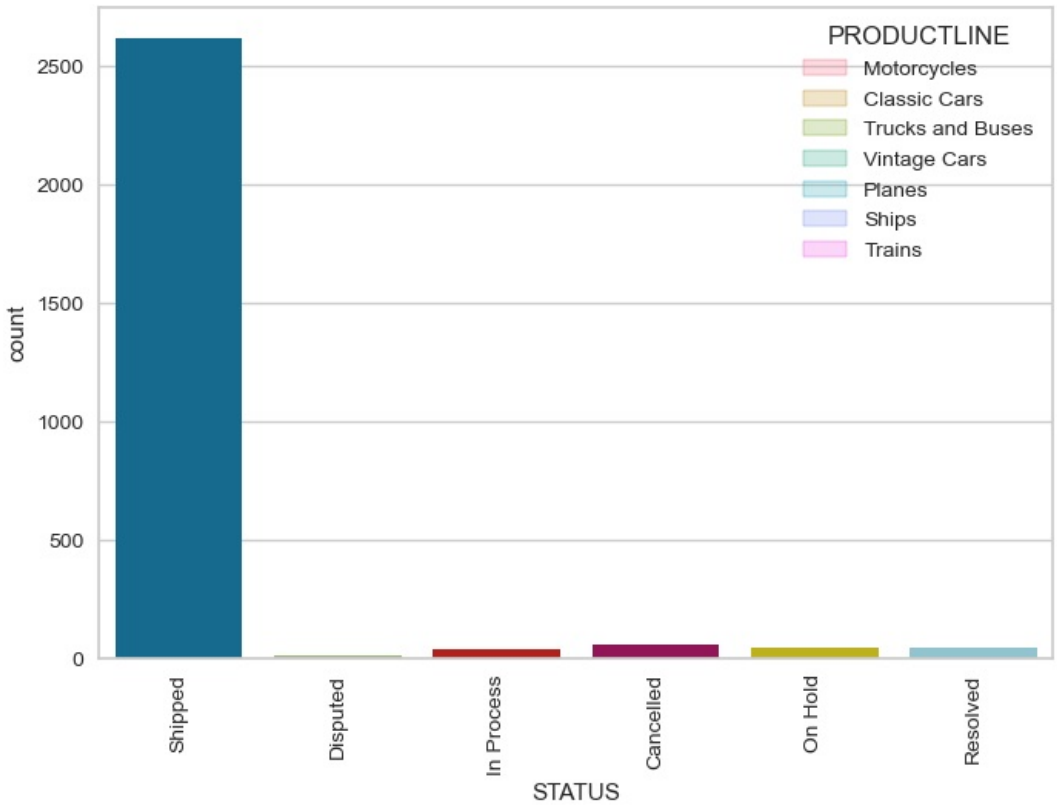
In [8]:
```
data.drop(["ORDERNUMBER", "PRICEEACH", "ORDERDATE", "PHONE", "ADDRESSLINE1", "ADDRESSLINE2", "CITY", "STATE", "
print(data.head() )
```

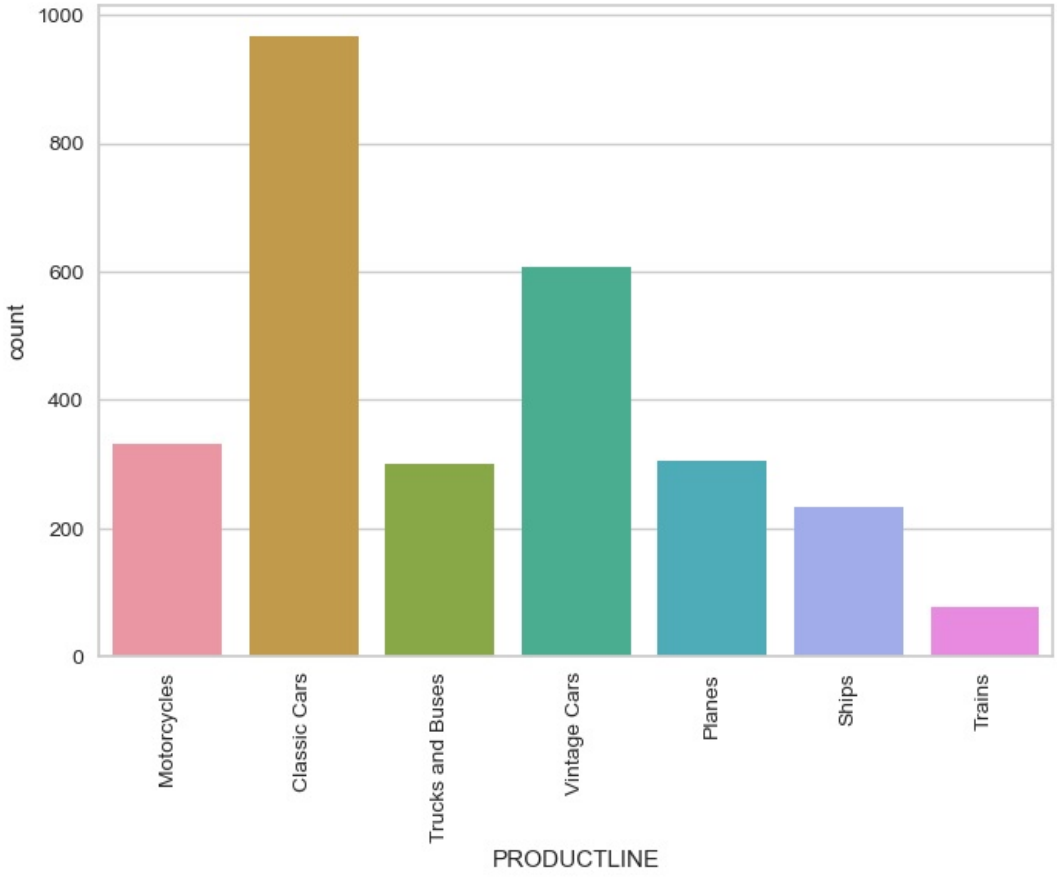
```
print(data.isnull().sum() )
print(data.describe() )
```

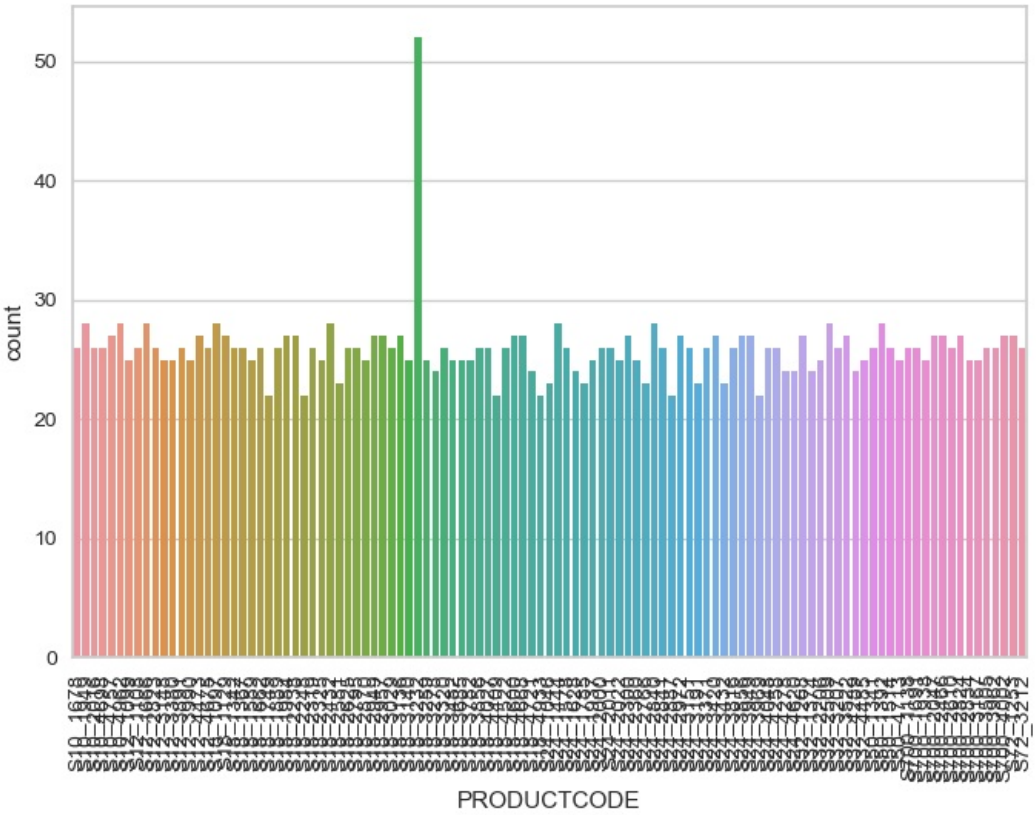
```
   QUANTITYORDERED  ORDERLINENUMBER    SALES   STATUS  QTR_ID  MONTH_ID  \
0               30                2  2871.00  Shipped       1         2
1               34                5  2765.90  Shipped       2         5
2               41                2  3884.34  Shipped       3         7
3               45                6  3746.70  Shipped       3         8
4               49               14  5205.27  Shipped       4        10

   YEAR_ID  PRODUCTLINE  MSRP PRODUCTCODE              CUSTOMERNAME COUNTRY  \
0     2003  Motorcycles    95    S10_1678         Land of Toys Inc.     USA
1     2003  Motorcycles    95    S10_1678        Reims Collectables  France
2     2003  Motorcycles    95    S10_1678           Lyon Souveniers  France
3     2003  Motorcycles    95    S10_1678         Toys4GrownUps.com     USA
4     2003  Motorcycles    95    S10_1678  Corporate Gift Ideas Co.     USA

  DEALSIZE
0    Small
1    Small
2   Medium
3   Medium
4   Medium
QUANTITYORDERED    0
ORDERLINENUMBER    0
SALES              0
STATUS             0
QTR_ID             0
MONTH_ID           0
YEAR_ID            0
PRODUCTLINE        0
MSRP               0
PRODUCTCODE        0
CUSTOMERNAME       0
COUNTRY            0
DEALSIZE           0
dtype: int64
       QUANTITYORDERED  ORDERLINENUMBER         SALES       QTR_ID  \
count      2823.000000      2823.000000   2823.000000  2823.000000
mean         35.092809         6.466171   3553.889072     2.717676
std           9.741443         4.225841   1841.865106     1.203878
min           6.000000         1.000000    482.130000     1.000000
25%          27.000000         3.000000   2203.430000     2.000000
50%          35.000000         6.000000   3184.800000     3.000000
75%          43.000000         9.000000   4508.000000     4.000000
max          97.000000        18.000000  14082.800000     4.000000

          MONTH_ID     YEAR_ID         MSRP
count  2823.000000  2823.00000  2823.000000
mean      7.092455  2003.81509   100.715551
std       3.656633     0.69967    40.187912
min       1.000000  2003.00000    33.000000
25%       4.000000  2003.00000    68.000000
50%       8.000000  2004.00000    99.000000
75%      11.000000  2004.00000   124.000000
max      12.000000  2005.00000   214.000000
```

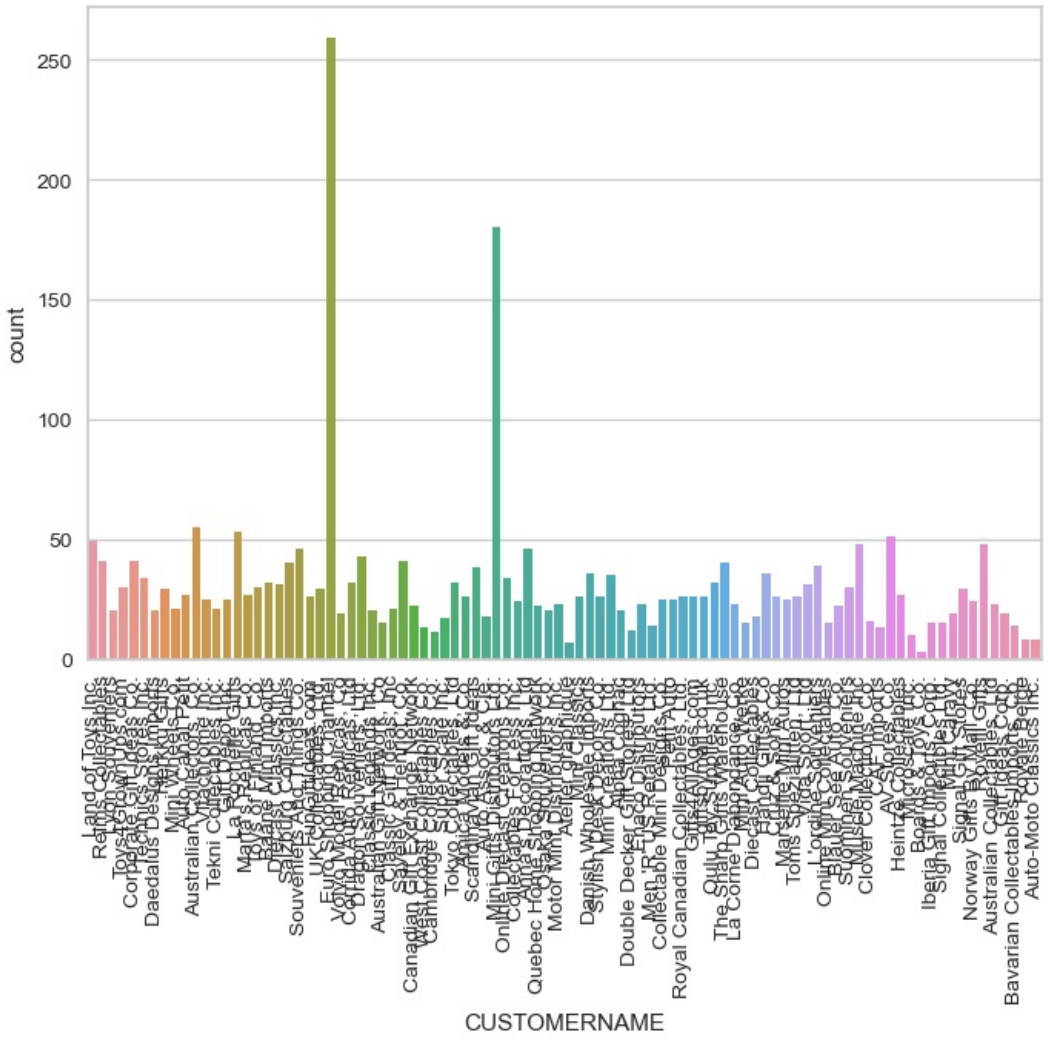
In [9]:
```
sns.countplot(data = data , x = 'STATUS')
sns.histplot(x = 'SALES' , hue = 'PRODUCTLINE', data = data,element="poly")
data['PRODUCTLINE'].unique()
data.drop_duplicates(inplace=True)
data.info()
list_cat = data.select_dtypes(include=['object']).columns.tolist()
list_cat
for i in list_cat:
    sns.countplot(data = data ,x = i)
    plt.xticks(rotation = 90)
    plt.show()
```

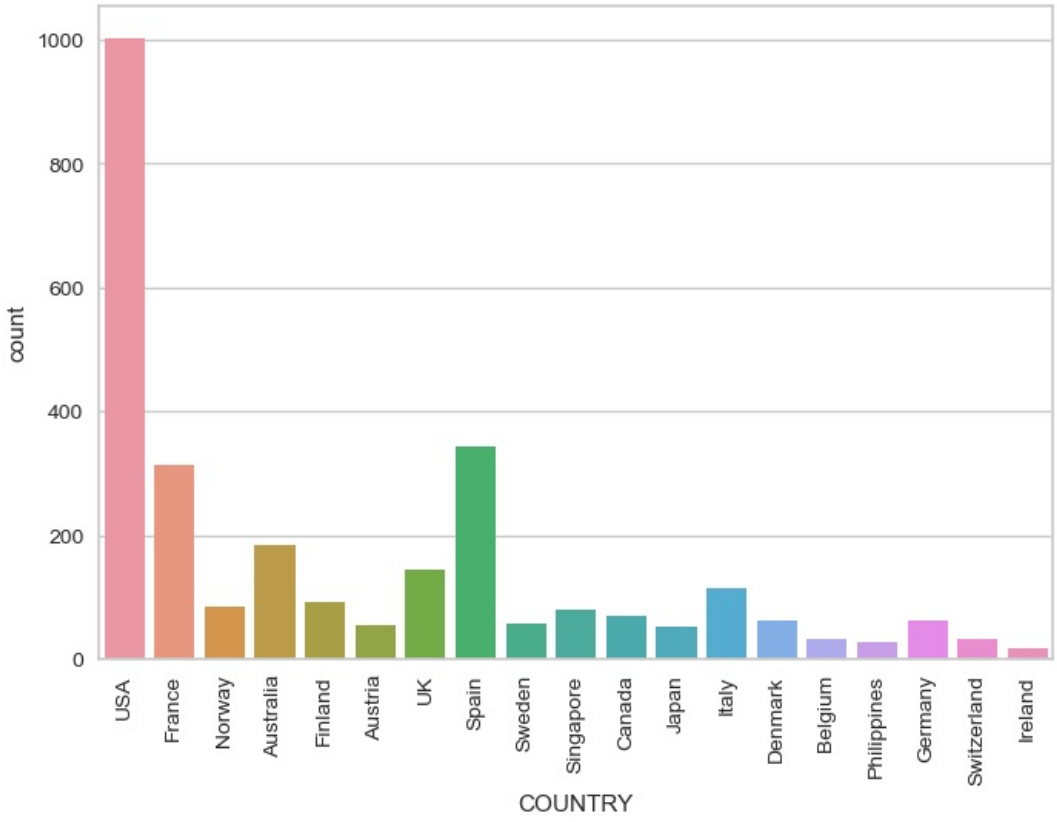
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   QUANTITYORDERED  2823 non-null   int64
 1   ORDERLINENUMBER  2823 non-null   int64
 2   SALES            2823 non-null   float64
 3   STATUS           2823 non-null   object
 4   QTR_ID           2823 non-null   int64
 5   MONTH_ID         2823 non-null   int64
 6   YEAR_ID          2823 non-null   int64
 7   PRODUCTLINE      2823 non-null   object
 8   MSRP             2823 non-null   int64
 9   PRODUCTCODE      2823 non-null   object
 10  CUSTOMERNAME     2823 non-null   object
 11  COUNTRY          2823 non-null   object
 12  DEALSIZE         2823 non-null   object
dtypes: float64(1), int64(6), object(6)
memory usage: 286.8+ KB
```

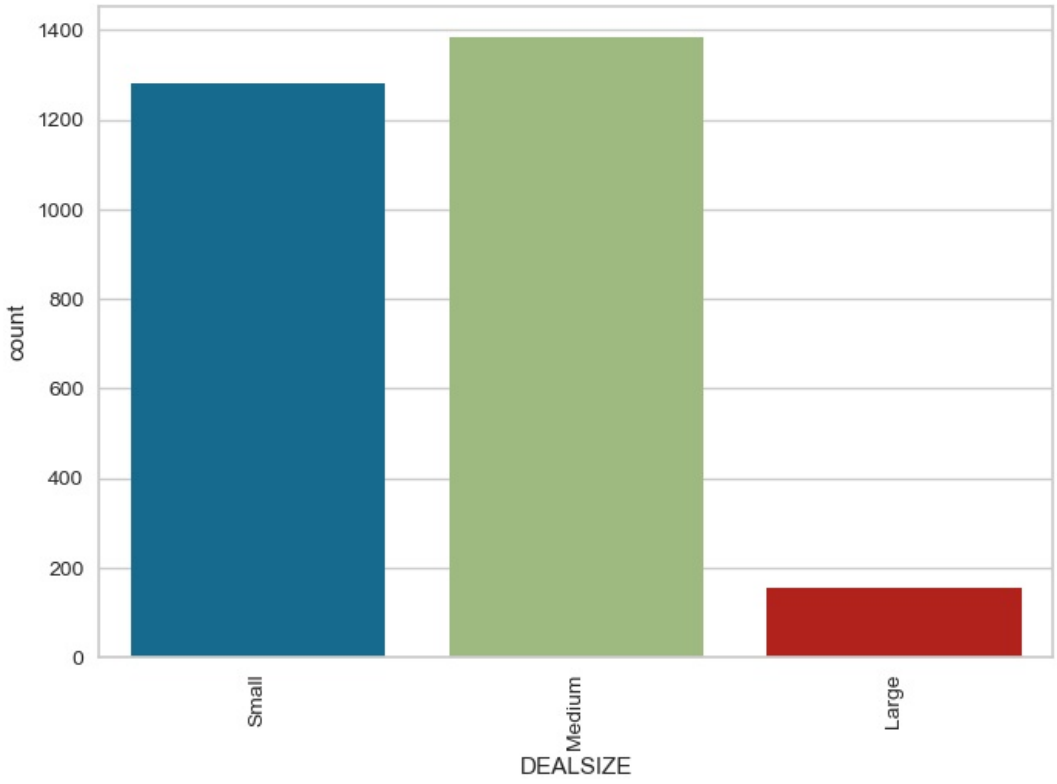

PRODUCTLINE
Motorcycles
Classic Cars
Trucks and Buses
Vintage Cars
Planes
Ships
Trains
count
2500
2000
1500
1000
500
0
Shipped
Disputed
In Process
Cancelled
On Hold
Resolved
STATUS


count
1000
800
600
400
200
0
Motorcycles
Classic Cars
Trucks and Buses
Vintage Cars
Planes
Ships
Trains
PRODUCTLINE


50
40
30
20
10
0
count
PRODUCTCODE


count
250
200
150
100
50
0
CUSTOMERNAME





In [10]:
```
le = preprocessing.LabelEncoder()
```

In [11]:
```
for i in list_cat:
    data[i]= le.fit_transform(data[i])
```

In [12]:
```
data.info()
data['SALES'] = data['SALES'].astype(int)
data.info()
data.describe()
X = data[['SALES','PRODUCTCODE']]
```

```
data.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   QUANTITYORDERED  2823 non-null   int64  
 1   ORDERLINENUMBER  2823 non-null   int64  
 2   SALES            2823 non-null   float64
 3   STATUS           2823 non-null   int32  
 4   QTR_ID           2823 non-null   int64  
 5   MONTH_ID         2823 non-null   int64  
 6   YEAR_ID          2823 non-null   int64  
 7   PRODUCTLINE      2823 non-null   int32  
 8   MSRP             2823 non-null   int64  
 9   PRODUCTCODE      2823 non-null   int32  
 10  CUSTOMERNAME     2823 non-null   int32  
 11  COUNTRY          2823 non-null   int32  
 12  DEALSIZE         2823 non-null   int32  
dtypes: float64(1), int32(6), int64(6)
memory usage: 220.7 KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   QUANTITYORDERED  2823 non-null   int64
 1   ORDERLINENUMBER  2823 non-null   int64
 2   SALES            2823 non-null   int32
 3   STATUS           2823 non-null   int32
 4   QTR_ID           2823 non-null   int64
 5   MONTH_ID         2823 non-null   int64
 6   YEAR_ID          2823 non-null   int64
 7   PRODUCTLINE      2823 non-null   int32
 8   MSRP             2823 non-null   int64
 9   PRODUCTCODE      2823 non-null   int32
 10  CUSTOMERNAME     2823 non-null   int32
 11  COUNTRY          2823 non-null   int32
 12  DEALSIZE         2823 non-null   int32
dtypes: int32(7), int64(6)
memory usage: 209.6 KB
```

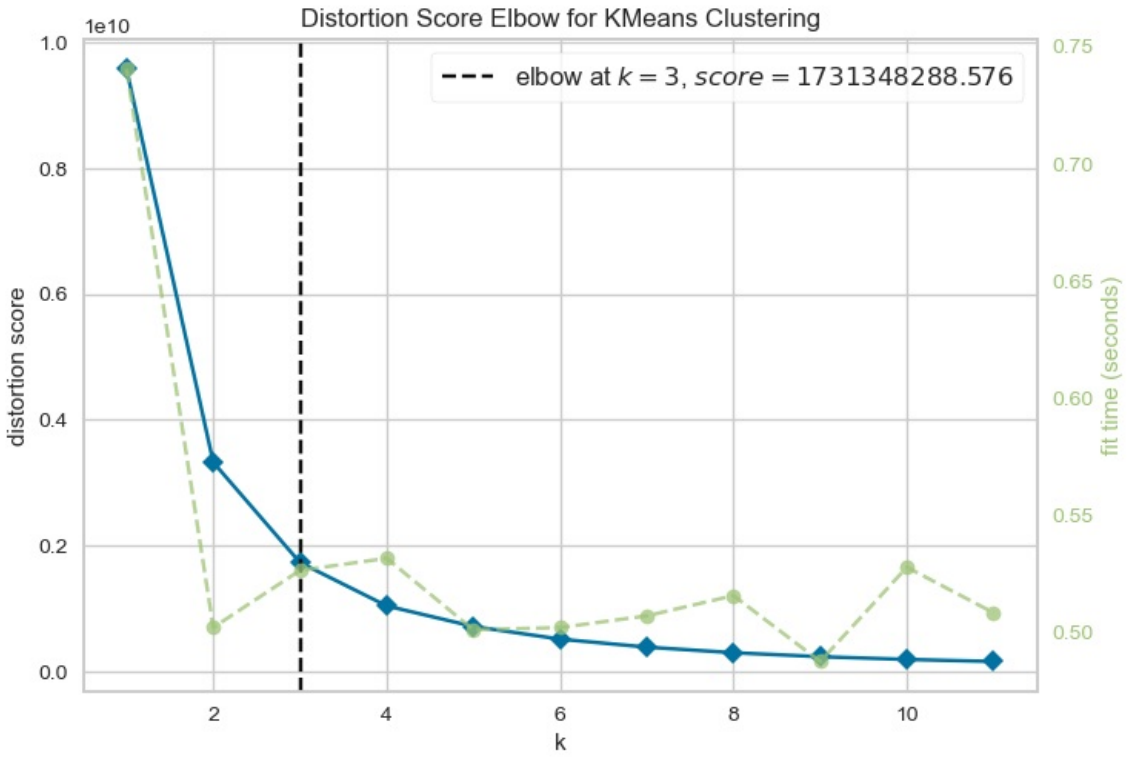
Out[12]:

```
Index(['QUANTITYORDERED', 'ORDERLINENUMBER', 'SALES', 'STATUS', 'QTR_ID',
       'MONTH_ID', 'YEAR_ID', 'PRODUCTLINE', 'MSRP', 'PRODUCTCODE',
       'CUSTOMERNAME', 'COUNTRY', 'DEALSIZE'],
      dtype='object')
```

In [13]:

```
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,12)).fit(X)
visualizer.show()
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=0).fit(X)
kmeans.labels_
```

```
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```




```
C:\Users\SHUBHAM\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value
of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

Out[13]:
```
array([3, 3, 3, ..., 1, 0, 3])
```

In [14]:
```
kmeans.inertia_
```

Out[14]:
```
1042124306.212494
```

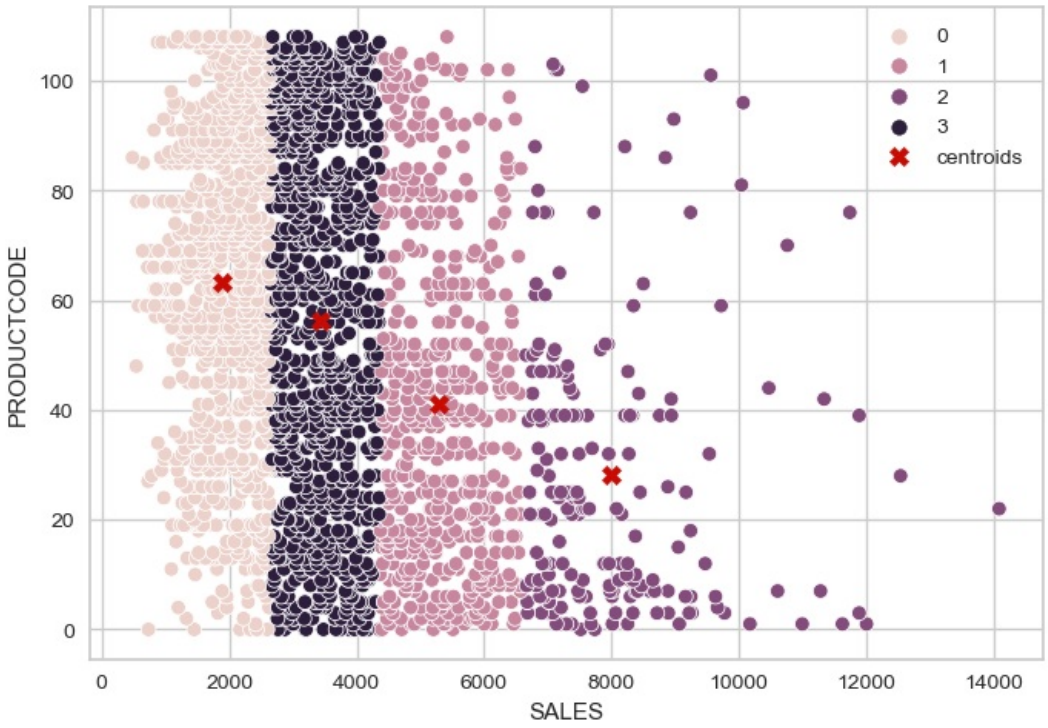
In [15]:
```
kmeans.n_iter_
kmeans.cluster_centers_
```

Out[15]:
```
array([[1882.98554913,   63.28420039],
       [5295.90973451,   40.97522124],
       [7983.1758794 ,   28.05025126],
       [3424.0244858 ,   56.19980411]])
```

In [16]:
```
Counter(kmeans.labels_)
```

Out[16]:
```
Counter({0: 1038, 3: 1023, 1: 563, 2: 199})
```

In [17]:
```
sns.scatterplot(data=X, x="SALES", y="PRODUCTCODE", hue=kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],marker="X", c="r", s=80, label="centroid
plt.legend()
plt.show()
```



In [ ]: