Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method.

In [ ]:

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

In [ ]:

```
df = pd.read_csv("sales_data_sample.csv", encoding = 'latin-1')
df
```

In [ ]:

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

In [ ]:

```
df.shape
```

In [ ]:

```
df.drop(['ADDRESSLINE1',"ADDRESSLINE2","STATE","TERRITORY", "CONTACTFIRST
```

In [ ]:

```
df.drop(['PHONE', 'YEAR_ID', 'ORDERDATE'], axis = 1, inplace = True)
df.info()
```

In [ ]:

```
df.describe()
```

In [ ]:

```
df.drop('ORDERNUMBER', inplace = True, axis = 1)
```

In [ ]:

```
df.describe()
```

In [ ]:

```
df.shape
```

In [ ]:

```
cat_list = df.select_dtypes(include=object).columns.to_list()
```

In [ ]:

```
cat_list
```

In [ ]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [ ]:

```
df[cat_list].nunique()
```

In [ ]:

```
for i in cat_list:
    df[i] = le.fit_transform(df[i])
```

In [ ]:

```
X = df[['SALES','PRODUCTCODE']]
```

In [ ]:

```
X.head()
```

In [ ]:

```
import warnings

warnings.filterwarnings('ignore', category=FutureWarning, module='sklearn
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans

visualizer = KElbowVisualizer(KMeans(), k = (1,12)).fit(X)
visualizer.show()
```

In [ ]:

```
kmeans_model = KMeans(n_clusters = 3, init = "k-means++").fit(X)
```

In [ ]:

```
kmeans_model.labels_
```

In [ ]:

```
kmeans_model.cluster_centers_
```

In [ ]:

```
from collections import Counter
Counter(kmeans_model.labels_)
```

In [ ]:

```
sns.scatterplot(data = X, x = 'SALES', y = 'PRODUCTCODE',hue = kmeans_mod
plt.scatter(kmeans_model.cluster_centers_[:,0], kmeans_model.cluster_cent
plt.legend()
plt.show()
```

In [ ]:

```

```