**Implement k-means clustering hierarchical clustering on sales data sampling.csv data set.**

**sales\_data\_samples.csv dataset required**

**pip install numpy**

**pip install pandas**

**pip install matplotlib**

**pip install seaborn**

**pip install sklearn**

**pip install yellobrick**

**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**

**data = pd.read\_csv('sales\_data\_samples.csv/sales\_data\_sample.csv', encoding='Latin-1')**

**print(data)**

**print(data.shape)**

**print(data.isnull().sum() )**

**data.drop(["ORDERNUMBER", "PRICEEACH", "ORDERDATE", "PHONE", "ADDRESSLINE1", "ADDRESSLINE2", "CITY", "STATE", "TERRITORY", "POSTALCODE", "CONTACTLASTNAME", "CONTACTFIRSTNAME"], axis = 1, inplace=True)**

**print(data.head() )**

**print(data.isnull().sum() )**

**print(data.describe() )**

**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()**

**le = preprocessing.LabelEncoder()**

**for i in list\_cat:**

**data[i]= le.fit\_transform(data[i])**

**data.info()**

**data['SALES'] = data['SALES'].astype(int)**

**data.info()**

**data.describe()**

**X = data[['SALES','PRODUCTCODE']]**

**data.columns**

**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\_**

**kmeans.inertia\_**

**kmeans.n\_iter\_**

**kmeans.cluster\_centers\_**

**Counter(kmeans.labels\_)**

**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="centroids")**

**plt.legend()**

**plt.show()**