**ML4**

**import** pandas **as** pd

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**from** sklearn.cluster **import** KMeans, k\_means

**from** sklearn.decomposition **import** PCA

df **=** pd**.**read\_csv("sales\_data\_sample.csv")

df**.**head()

df**.**shape

df**.**describe()

df**.**info()

df**.**isnull()**.**sum()

df**.**dtypes

df\_drop **=** ['ADDRESSLINE1', 'ADDRESSLINE2', 'STATUS','POSTALCODE', 'CITY', 'TERRITORY', 'PHONE', 'STATE', 'CONTACTFIRSTNAME', 'CONTACTLASTNAME', 'CUSTOMERNAME', 'ORDERNUMBER']

df **=** df**.**drop(df\_drop, axis**=**1)

df**.**isnull()**.**sum()

df**.**dtypes

df['COUNTRY']**.**unique()

df['PRODUCTLINE']**.**unique()

df['DEALSIZE']**.**unique()

productline **=** pd**.**get\_dummies(df['PRODUCTLINE'])

Dealsize **=** pd**.**get\_dummies(df['DEALSIZE'])

df **=** pd**.**concat([df,productline,Dealsize], axis **=** 1)

df\_drop **=** ['COUNTRY','PRODUCTLINE','DEALSIZE']

df **=** df**.**drop(df\_drop, axis**=**1)

df['PRODUCTCODE'] **=** pd**.**Categorical(df['PRODUCTCODE'])**.**codes

df**.**drop('ORDERDATE', axis**=**1, inplace**=True**)

df**.**dtypes

distortions **=** []

K **=** range(1,10)

**for** k **in** K:

kmeanModel **=** KMeans(n\_clusters**=**k)

kmeanModel**.**fit(df)

distortions**.**append(kmeanModel**.**inertia\_)

plt**.**figure(figsize**=**(16,8))

plt**.**plot(K, distortions, 'bx-')

plt**.**xlabel('k')

plt**.**ylabel('Distortion')

plt**.**title('The Elbow Method showing the optimal k')

plt**.**show()

X\_train **=** df**.**values

X\_train**.**shape

model **=** KMeans(n\_clusters**=**3,random\_state**=**2)

model **=** model**.**fit(X\_train)

predictions **=** model**.**predict(X\_train)

unique,counts **=** np**.**unique(predictions,return\_counts**=True**)

counts **=** counts**.**reshape(1,3)

counts\_df **=** pd**.**DataFrame(counts,columns**=**['Cluster1','Cluster2','Cluster3'])

counts\_df**.**head()

pca **=** PCA(n\_components**=**2)

reduced\_X **=** pd**.**DataFrame(pca**.**fit\_transform(X\_train),columns**=**['PCA1','PCA2'])

reduced\_X**.**head()

plt**.**figure(figsize**=**(14,10))

plt**.**scatter(reduced\_X['PCA1'],reduced\_X['PCA2'])

model**.**cluster\_centers\_

reduced\_centers **=** pca**.**transform(model**.**cluster\_centers\_)

reduced\_centers

plt**.**figure(figsize**=**(14,10))

plt**.**scatter(reduced\_X['PCA1'],reduced\_X['PCA2'])

plt**.**scatter(reduced\_centers[:,0],reduced\_centers[:,1],color**=**'black',marker**=**'x',s**=**300)

reduced\_X['Clusters'] **=** predictions

reduced\_X**.**head()

plt**.**figure(figsize**=**(14,10))

plt**.**scatter(reduced\_X[reduced\_X['Clusters'] **==** 0]**.**loc[:,'PCA1'],reduced\_X[reduced\_X['Clusters'] **==** 0]**.**loc[:,'PCA2'],color**=**'slateblue')

plt**.**scatter(reduced\_X[reduced\_X['Clusters'] **==** 1]**.**loc[:,'PCA1'],reduced\_X[reduced\_X['Clusters'] **==** 1]**.**loc[:,'PCA2'],color**=**'springgreen')

plt**.**scatter(reduced\_X[reduced\_X['Clusters'] **==** 2]**.**loc[:,'PCA1'],reduced\_X[reduced\_X['Clusters'] **==** 2]**.**loc[:,'PCA2'],color**=**'indigo')

plt**.**scatter(reduced\_centers[:,0],reduced\_centers[:,1],color**=**'black',marker**=**'x',s**=**300)