

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
#Importing the required libraries.
```

```
In [2]: from sklearn.cluster import KMeans, k_means #For clustering
from sklearn.decomposition import PCA #Linear Dimensionality reduction.
```

```
In [4]: df = pd.read_csv(r"C:\Users\GAYATRI\Downloads\sales_data_sample.csv",encoding='Lat
```

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

```
Out[5]:
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE
0	10107	30	95.70	2	2871.00	2/24/2003 0:00
1	10121	34	81.35	5	2765.90	5/7/2003 0:00
2	10134	41	94.74	2	3884.34	7/1/2003 0:00
3	10145	45	83.26	6	3746.70	8/25/2003 0:00
4	10159	49	100.00	14	5205.27	10/10/2003 0:00

5 rows × 25 columns

```
In [6]: df.shape
```

```
Out[6]: (2823, 25)
```

```
In [7]: df.describe()
```

```
Out[7]:
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000
mean	10258.725115	35.092809	83.658544	6.466171	3553.889072
std	92.085478	9.741443	20.174277	4.225841	1841.865106
min	10100.000000	6.000000	26.880000	1.000000	482.130000
25%	10180.000000	27.000000	68.860000	3.000000	2203.430000
50%	10262.000000	35.000000	95.700000	6.000000	3184.800000
75%	10333.500000	43.000000	100.000000	9.000000	4508.000000
max	10425.000000	97.000000	100.000000	18.000000	14082.800000

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ORDERNUMBER           2823 non-null   int64
1   QUANTITYORDERED       2823 non-null   int64
2   PRICEEACH             2823 non-null   float64
3   ORDERLINENUMBER       2823 non-null   int64
4   SALES                 2823 non-null   float64
5   ORDERDATE             2823 non-null   object
6   STATUS                2823 non-null   object
7   QTR_ID                2823 non-null   int64
8   MONTH_ID              2823 non-null   int64
9   YEAR_ID               2823 non-null   int64
10  PRODUCTLINE           2823 non-null   object
11  MSRP                  2823 non-null   int64
12  PRODUCTCODE           2823 non-null   object
13  CUSTOMERNAME          2823 non-null   object
14  PHONE                 2823 non-null   object
15  ADDRESSLINE1          2823 non-null   object
16  ADDRESSLINE2          302 non-null    object
17  CITY                  2823 non-null   object
18  STATE                 1337 non-null   object
19  POSTALCODE            2747 non-null   object
20  COUNTRY               2823 non-null   object
21  TERRITORY             1749 non-null   object
22  CONTACTLASTNAME       2823 non-null   object
23  CONTACTFIRSTNAME      2823 non-null   object
24  DEALSIZE              2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: 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 [10]: df.dtypes
```

```
Out[10]: ORDERNUMBER          int64
          QUANTITYORDERED    int64
          PRICEEACH          float64
          ORDERLINENUMBER    int64
          SALES              float64
          ORDERDATE          object
          STATUS             object
          QTR_ID             int64
          MONTH_ID          int64
          YEAR_ID           int64
          PRODUCTLINE        object
          MSRP              int64
          PRODUCTCODE        object
          CUSTOMERNAME       object
          PHONE              object
          ADDRESSLINE1       object
          ADDRESSLINE2       object
          CITY               object
          STATE              object
          POSTALCODE         object
          COUNTRY            object
          TERRITORY          object
          CONTACTLASTNAME    object
          CONTACTFIRSTNAME   object
          DEALSIZE           object
          dtype: object
```

```
In [11]: df_drop = ['ADDRESSLINE1', 'ADDRESSLINE2', 'STATUS', 'POSTALCODE', 'CITY', 'TERRITORY']
          df = df.drop(df_drop, axis=1) #Dropping the categorical unnecessary columns along with
```

```
In [12]: df.isnull().sum()
```

```
Out[12]: QUANTITYORDERED    0
          PRICEEACH          0
          ORDERLINENUMBER    0
          SALES              0
          ORDERDATE          0
          QTR_ID             0
          MONTH_ID          0
          YEAR_ID           0
          PRODUCTLINE        0
          MSRP              0
          PRODUCTCODE        0
          COUNTRY            0
          DEALSIZE           0
          dtype: int64
```

```
In [13]: df.dtypes
```

```
Out[13]: QUANTITYORDERED    int64
          PRICEEACH          float64
          ORDERLINENUMBER    int64
          SALES              float64
          ORDERDATE          object
          QTR_ID             int64
          MONTH_ID          int64
          YEAR_ID           int64
          PRODUCTLINE        object
          MSRP              int64
          PRODUCTCODE        object
          COUNTRY            object
          DEALSIZE           object
          dtype: object
```

```

In [14]: df['COUNTRY'].unique()

Out[14]: array(['USA', 'France', 'Norway', 'Australia', 'Finland', 'Austria', 'UK',
        'Spain', 'Sweden', 'Singapore', 'Canada', 'Japan', 'Italy',
        'Denmark', 'Belgium', 'Philippines', 'Germany', 'Switzerland',
        'Ireland'], dtype=object)

In [15]: df['PRODUCTLINE'].unique()

Out[15]: array(['Motorcycles', 'Classic Cars', 'Trucks and Buses', 'Vintage Cars',
        'Planes', 'Ships', 'Trains'], dtype=object)

In [16]: df['DEALSIZE'].unique()

Out[16]: array(['Small', 'Medium', 'Large'], dtype=object)

In [17]: productline = pd.get_dummies(df['PRODUCTLINE']) #Converting the categorical columns
Dealsize = pd.get_dummies(df['DEALSIZE'])

In [18]: df = pd.concat([df,productline,Dealsize], axis = 1)

In [19]: df_drop = ['COUNTRY','PRODUCTLINE','DEALSIZE'] #Dropping Country too as there are
df = df.drop(df_drop, axis=1)

In [20]: df['PRODUCTCODE'] = pd.Categorical(df['PRODUCTCODE']).codes #Converting the datatype

In [21]: df.drop('ORDERDATE', axis=1, inplace=True) #Dropping the Orderdate as Month is already there

In [22]: df.dtypes #All the datatypes are converted into numeric

Out[22]: QUANTITYORDERED      int64
PRICEEACH                    float64
ORDERLINENUMBER              int64
SALES                        float64
QTR_ID                       int64
MONTH_ID                     int64
YEAR_ID                      int64
MSRP                         int64
PRODUCTCODE                   int8
Classic Cars                  uint8
Motorcycles                   uint8
Planes                        uint8
Ships                         uint8
Trains                        uint8
Trucks and Buses              uint8
Vintage Cars                  uint8
Large                         uint8
Medium                        uint8
Small                         uint8
dtype: object

In [23]: distortions = [] # Within Cluster Sum of Squares from the centroid
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df)
    distortions.append(kmeanModel.inertia_) #Appending the inertia to the Distortions

```

```

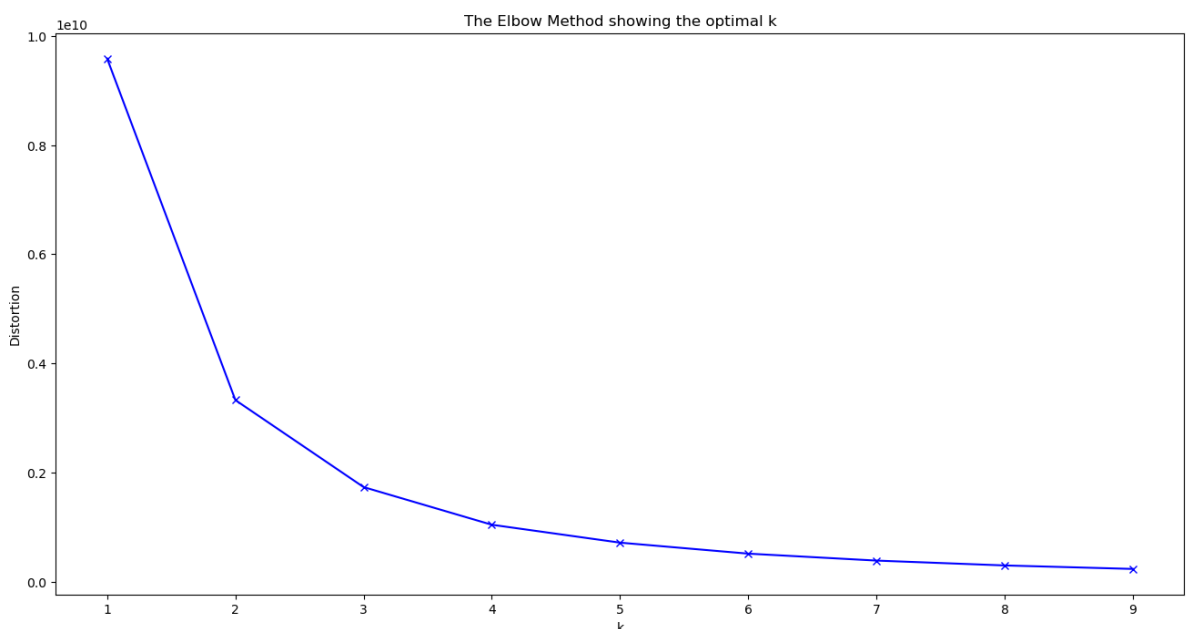
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(
C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\_kmeans.py:870: 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
  warnings.warn(

```

```

In [48]: 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()

```



```
In [25]: X_train = df.values #Returns a numpy array.
```

```
In [26]: X_train.shape
```

```
Out[26]: (2823, 19)
```

```
In [27]: model = KMeans(n_clusters=3,random_state=2) #Number of cluster = 3
model = model.fit(X_train) #Fitting the values to create a model.
predictions = model.predict(X_train) #Predicting the cluster values (0,1,or 2)
```

C:\Users\GAYATRI\anaconda31\lib\site-packages\sklearn\cluster\\_kmeans.py:870: 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  
warnings.warn(

```
In [28]: unique_counts = np.unique(predictions,return_counts=True)
```

```
In [29]: counts = counts.reshape(1,3)
```

```
In [30]: counts_df = pd.DataFrame(counts,columns=['Cluster1','Cluster2','Cluster3'])
```

```
In [31]: counts_df.head()
```

```
Out[31]:
```

	Cluster1	Cluster2	Cluster3
0	1083	1367	373

```
In [32]: pca = PCA(n_components=2) #Converting all the features into 2 columns to make it ec
```

```
In [33]: reduced_X = pd.DataFrame(pca.fit_transform(X_train),columns=['PCA1','PCA2']) #Creat
```

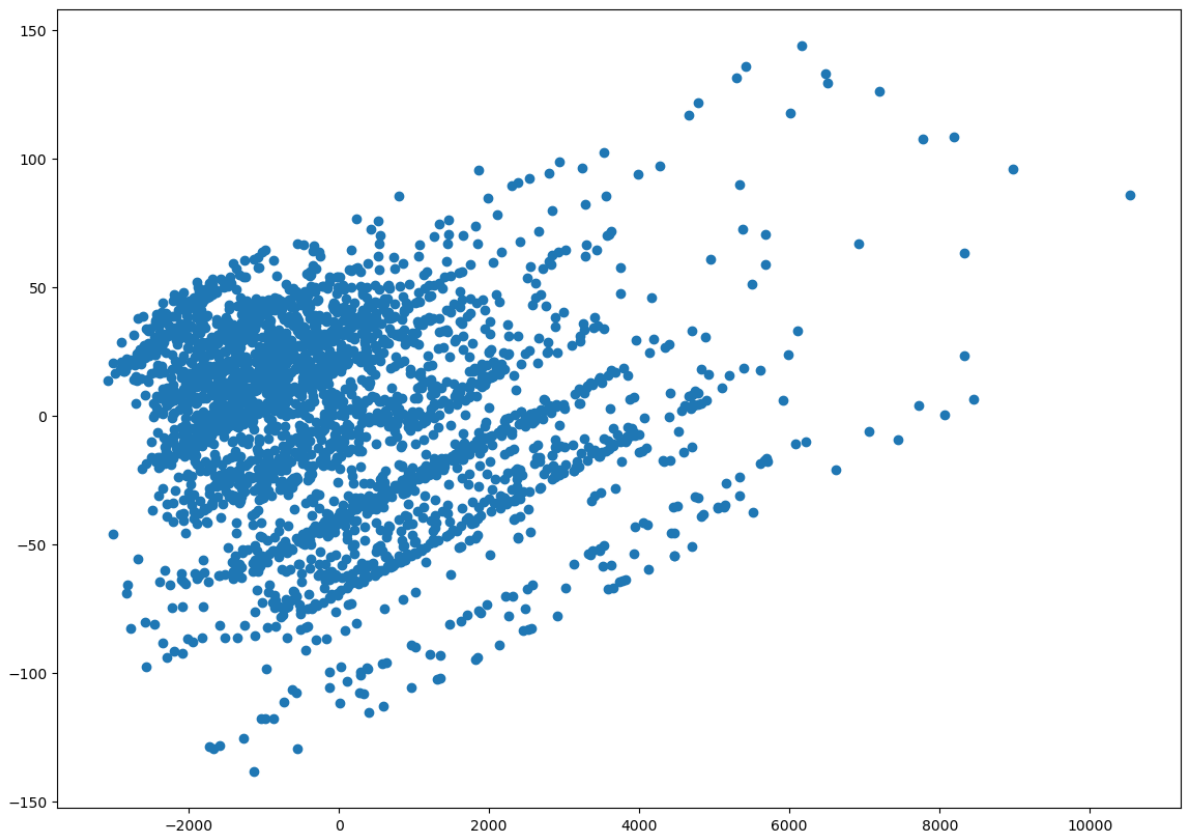
```
In [34]: reduced_X.head()
```

```
Out[34]:
```

	PCA1	PCA2
0	-682.488323	-42.819535
1	-787.665502	-41.694991
2	330.732170	-26.481208
3	193.040232	-26.285766
4	1651.532874	-6.891196

```
In [35]: #Plotting the normal Scatter Plot
plt.figure(figsize=(14,10))
plt.scatter(reduced_X['PCA1'],reduced_X['PCA2'])
```

```
Out[35]: <matplotlib.collections.PathCollection at 0x1f6a04a1ba0>
```



In [36]: `model.cluster_centers_` *#Finding the centriods. (3 Centriods in total. Each Array co*

Out[36]: `array([[ 3.72031394e+01, 9.52120960e+01, 6.44967682e+00, 4.13868425e+03, 2.72022161e+00, 7.09879963e+00, 2.00379409e+03, 1.13248384e+02, 5.04469067e+01, 3.74884580e-01, 1.15420129e-01, 9.41828255e-02, 8.21791320e-02, 1.84672207e-02, 1.16343490e-01, 1.98522622e-01, 2.08166817e-17, 1.00000000e+00, -6.66133815e-16],  
 [ 3.08302853e+01, 7.00755230e+01, 6.67300658e+00, 2.12409474e+03, 2.71762985e+00, 7.09509876e+00, 2.00381127e+03, 7.84784199e+01, 6.24871982e+01, 2.64813460e-01, 1.21433797e-01, 1.29480614e-01, 1.00219459e-01, 3.87710315e-02, 9.21726408e-02, 2.53108998e-01, 2.08166817e-17, 6.21799561e-02, 9.37820044e-01],  
 [ 4.45871314e+01, 9.98931099e+01, 5.75603217e+00, 7.09596863e+03, 2.71045576e+00, 7.06434316e+00, 2.00389008e+03, 1.45823056e+02, 3.14959786e+01, 5.33512064e-01, 1.07238606e-01, 7.23860590e-02, 2.14477212e-02, 1.07238606e-02, 1.31367292e-01, 1.23324397e-01, 4.20911528e-01, 5.79088472e-01, 5.55111512e-17]])`

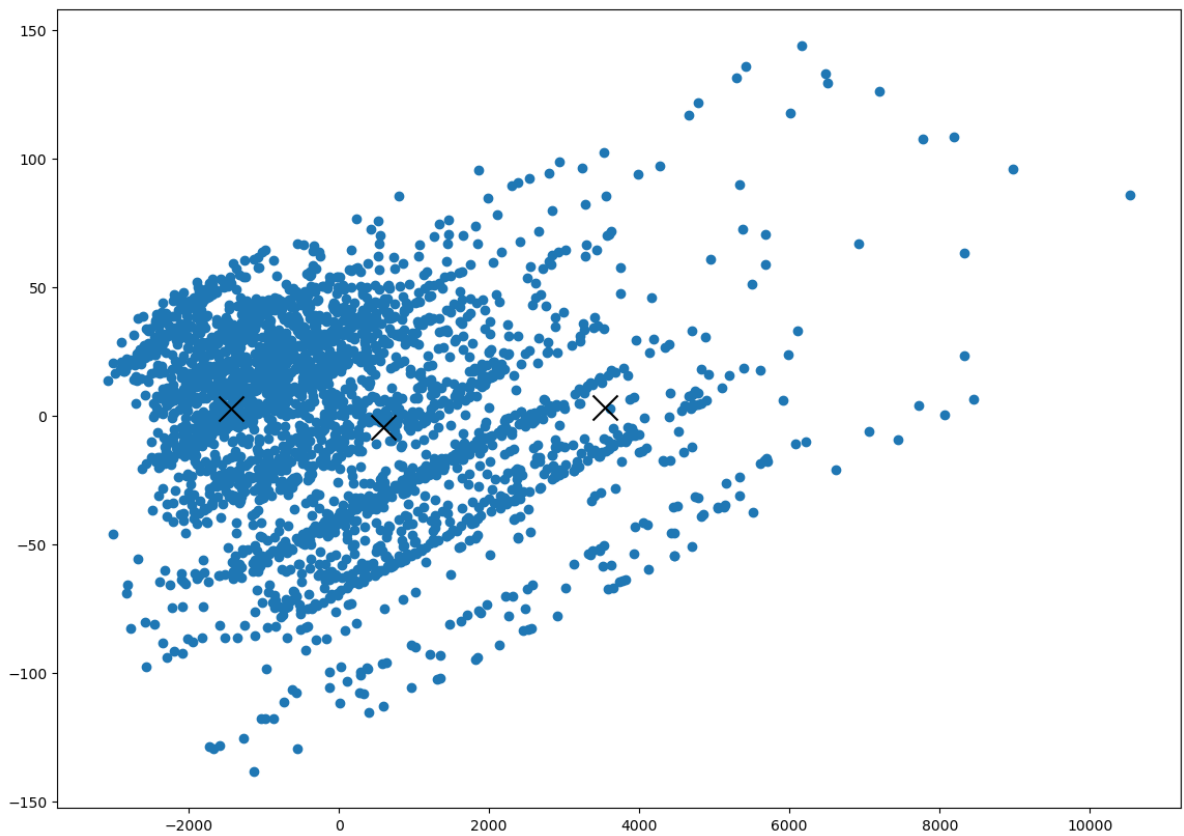
In [37]: `reduced_centers = pca.transform(model.cluster_centers_)` *#Transforming the centroids*

In [38]: `reduced_centers`

Out[38]: `array([[ 5.84994044e+02, -4.36786931e+00],  
 [-1.43005891e+03, 2.60041009e+00],  
 [ 3.54247180e+03, 3.15185487e+00]])`

In [39]: `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)`

Out[39]: `<matplotlib.collections.PathCollection at 0x1f6a2cf28c0>`



In [40]: `reduced_X['Clusters'] = predictions` *#Adding the Clusters to the reduced dataframe.*

In [41]: `reduced_X.head()`

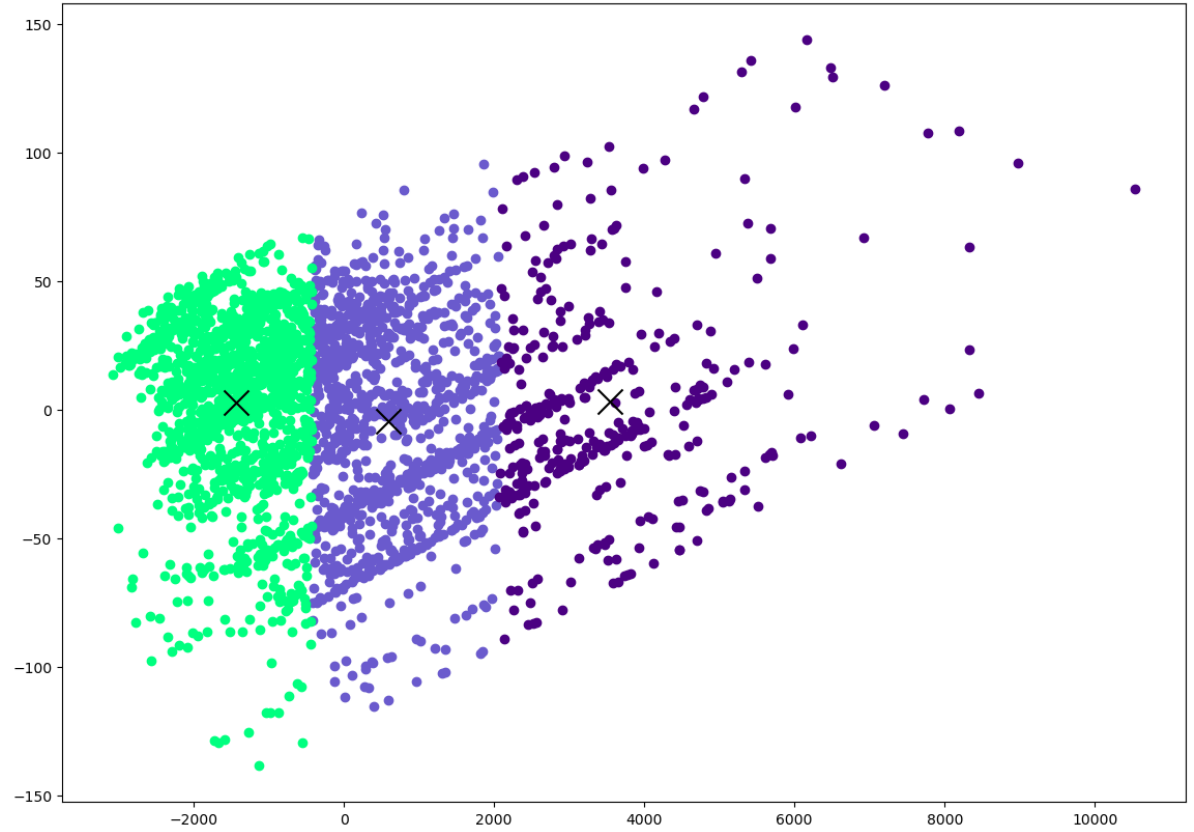
Out[41]:

	PCA1	PCA2	Clusters
0	-682.488323	-42.819535	1
1	-787.665502	-41.694991	1
2	330.732170	-26.481208	0
3	193.040232	-26.285766	0
4	1651.532874	-6.891196	0

In [42]: *#Plotting the clusters*  
`plt.figure(figsize=(14,10))`  
*# taking the cluster number and first column taking the*  
`plt.scatter(reduced_X[reduced_X['Clusters'] == 0].loc[:, 'PCA1'], reduced_X[reduced_X['Clusters'] == 0].loc[:, 'PCA2'], color='red', marker='o', s=30)`  
`plt.scatter(reduced_X[reduced_X['Clusters'] == 1].loc[:, 'PCA1'], reduced_X[reduced_X['Clusters'] == 1].loc[:, 'PCA2'], color='green', marker='o', s=30)`  
`plt.scatter(reduced_X[reduced_X['Clusters'] == 2].loc[:, 'PCA1'], reduced_X[reduced_X['Clusters'] == 2].loc[:, 'PCA2'], color='blue', marker='o', s=30)`  
`plt.scatter(reduced_centers[:,0], reduced_centers[:,1], color='black', marker='x', s=30)`

Out[42]: `<matplotlib.collections.PathCollection at 0x1f6a2d9a2f0>`





In [ ]:

In [ ]: