ML LAB ASSIGNMENT 2

```
# For dataframe handling and pandas operations
import pandas as pd

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import zipfile
import cv2
import plotly.express as px


from sklearn.preprocessing import StandardScaler, normalize
from sklearn.cluster import KMeans

%matplotlib inline


df = pd.read_csv('./sales_data_sample.csv', encoding = 'unicode_escape', parse_dates=['ORDERDATE'])



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

```
QUANTITYORDERED     0
PRICEEACH           0
ORDERLINENUMBER     0
SALES               0
MONTH_ID            0
YEAR_ID             0
MSRP                0
PRODUCTCODE         0
Australia           0
Austria             0
Belgium             0
Canada              0
Denmark             0
Finland             0
France              0
Germany             0
Ireland             0
Italy               0
Japan               0
Norway              0
Philippines         0
Singapore           0
Spain               0
Sweden              0
Switzerland         0
UK                  0
USA                 0
Classic Cars        0
Motorcycles         0
Planes              0
Ships               0
Trains              0
Trucks and Buses    0
Vintage Cars        0
Large               0
Medium              0
Small               0
dtype: int64
```

```
from google.colab import drive
drive.mount('/content/drive')


df_drop  = ['ADDRESSLINE1', 'ADDRESSLINE2', 'POSTALCODE', 'CITY', 'TERRITORY', 'PHONE', 'STATE', 'CONTACTFIRSTNAME', 'CONTACTLASTNAME', '
df = df.drop(df_drop, axis=1)
df.head(3)
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	M
0	30	95.70	2	2871.00	2003-02-24	Shipped	1	
1	34	81.35	5	2765.90	2003-05-07	Shipped	2	

▾ Drop georaphic features and names, phone

```
df.shape
```

```
(2823, 14)
```

```
df.isna().sum()
```

```
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
COUNTRY            0
DEALSIZE           0
dtype: int64
```

Drop unbalanced feature

```
df.drop(columns=['STATUS'], axis=1, inplace=True)
```

```
print('Columns resume: ', df.shape[1])
```

```
Columns resume:  13
```

Prepare data

```
def dummies(x):
  dummy = pd.get_dummies(df[x])
  df.drop(columns=x, inplace=True)
  return pd.concat([df, dummy], axis = 1)
```

```
df =  dummies('COUNTRY')
```

```
df =  dummies('PRODUCTLINE')
```

```
df =  dummies('DEALSIZE')
```

```
df.head()
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	QTR_ID	MONTH_ID	YEAR_ID	MSRP	PRODUCTCODE	...	Classic Cars	Motor
0	30	95.70	2	2871.00	2003-02-24	1	2	2003	95	S10_1678	...	False	
1	34	81.35	5	2765.90	2003-05-07	2	5	2003	95	S10_1678	...	False	
2	41	94.74	2	3884.34	2003-07-01	3	7	2003	95	S10_1678	...	False	
3	45	83.26	6	3746.70	2003-08-25	3	8	2003	95	S10_1678	...	False	
4	49	100.00	14	5205.27	2003-10-10	4	10	2003	95	S10_1678	...	False	

5 rows × 39 columns

```
y = pd.Categorical(df['PRODUCTCODE'])
y
```

```
['S10_1678', 'S10_1678', 'S10_1678', 'S10_1678', 'S10_1678', ..., 'S72_3212', 'S72_3212', 'S72_3212', 'S72_3212', 'S72_3212']
Length: 2823
Categories (109, object): ['S10_1678', 'S10_1949', 'S10_2016', 'S10_4698', ..., 'S700_3962', 'S700_4002', 'S72_1253', 'S72_3212']
```

Double-click (or enter) to edit

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

```
df.head()
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	QTR_ID	MONTH_ID
0	30	95.70	2	2871.00	2003-02-24	1	2
1	34	81.35	5	2765.90	2003-05-07	2	5
2	41	94.74	2	3884.34	2003-07-01	3	7
3	45	83.26	6	3746.70	2003-08-25	3	8
4	49	100.00	14	5205.27	2003-10-10	4	10

5 rows × 39 columns

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

drop 'ORDERDATE', 'QTR_ID' because we have 'MONTH' etc.

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

```
df.shape
```

```
(2823, 37)
```

Use K-MEANS algorithm

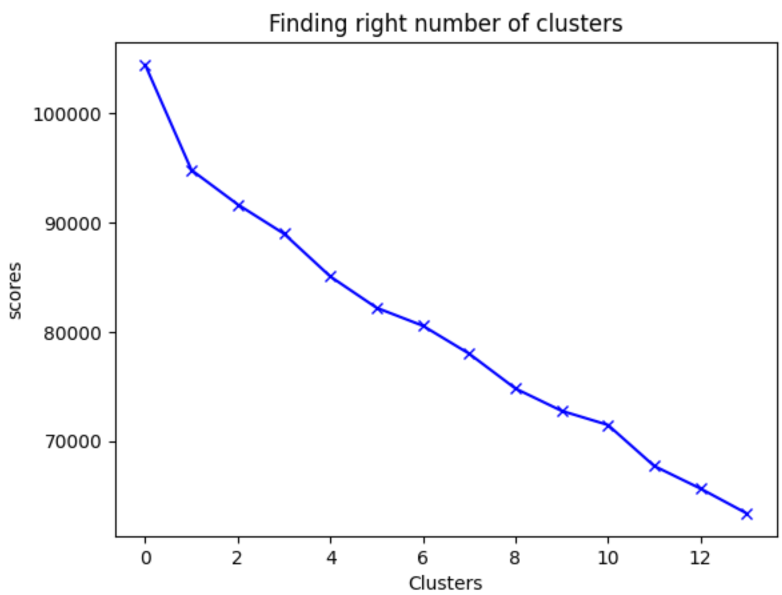
```
scaler =  StandardScaler()
df_scaled = scaler.fit_transform(df)
```

```
scores = []
range_values = range(1, 15)
for i in range_values:
  kmeans = KMeans(n_clusters = i)
  kmeans.fit(df_scaled)
  scores.append(kmeans.inertia_)
```

```
C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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
```

```
C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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
```

```
plt.plot(scores, 'bx-')
plt.title('Finding right number of clusters')
plt.xlabel('Clusters')
plt.ylabel('scores')
plt.show();
```



The elbow method

```
kmeans = KMeans(4)
kmeans.fit(df_scaled)
```

```
C:\Users\meenu\anaconda32\envs\tf\lib\site-packages\sklearn\cluster\_kmeans.py:1416: 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

▾ KMeans
KMeans(n_clusters=4)
```

```
labels = kmeans.labels_
labels
```

```
array([0, 0, 1, ..., 1, 0, 1])
```

```
kmeans.cluster_centers_.shape
```

```
(4, 37)
```

```
cluster_centers = pd.DataFrame(data = kmeans.cluster_centers_, columns= [df.columns])
cluster_centers
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	MONTH_ID	YEAR_ID	MSRP
0	-0.465084	-0.726204	0.039829	-0.810054	0.005971	-0.001150	-0.5858
1	0.299596	0.574416	-0.004393	0.457362	-0.004848	-0.015404	0.3740
2	1.245428	0.800220	-0.259579	2.573861	-0.088008	0.136857	1.4302
3	-0.173920	-0.039573	-0.005290	-0.189818	0.073057	-0.000691	-0.0723

4 rows × 37 columns

Invert the data

```
cluster_centers = scaler.inverse_transform(cluster_centers)
cluster_centers = pd.DataFrame(data=cluster_centers, columns=[df.columns])
cluster_centers
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	MONTH_ID	YEAR_ID	MS
0	30.563025	69.010496	6.634454	2062.143941	7.114286	2003.814286	7
1	38.010786	95.244923	6.447612	4396.138089	7.074730	2003.804314	11
2	47.222930	99.799554	5.369427	8293.753248	6.770701	2003.910828	15
3	33.398876	82.860337	6.443820	3204.331798	7.359551	2003.814607	9

4 rows × 37 columns

```
sales_of_cluster = pd.concat([df, pd.DataFrame({'cluster': labels})], axis=1)
sales_of_cluster.head()
```

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	MONTH_ID	YEAR_ID	MSRP	PROD
0	30	95.70	2	2871.00	2	2003	95	
1	34	81.35	5	2765.90	5	2003	95	
2	41	94.74	2	3884.34	7	2003	95	
3	45	83.26	6	3746.70	8	2003	95	
4	49	100.00	14	5205.27	10	2003	95	

5 rows × 38 columns