Machine Learning Laboratory – Assignment 6

- **NAME :- ANURAG AVINASH SHEVALE**
- **CLASS :- BE COMP I**
- **ROLL NO :- 20**

In [ ]:

```
#Name - Anurag Avinash Shevale
#Class - BE Comp I
#Roll No - 20
```

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

In [4]:

```
data = pd.read_csv('/home/admin1/Anurag/sales.csv',encoding='unicode_escape')
```

In [6]:

```
data.head(10)
```

Out[6]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDER
0	10107	30	95.70	2	2871.00	2/2
1	10121	34	81.35	5	2765.90	5/7/200
2	10134	41	94.74	2	3884.34	7/1/200
3	10145	45	83.26	6	3746.70	8/2
4	10159	49	100.00	14	5205.27	10/1
5	10168	36	96.66	1	3479.76	10/2
6	10180	29	86.13	9	2497.77	11/1
7	10188	48	100.00	1	5512.32	11/1
8	10201	22	98.57	2	2168.54	12/
9	10211	41	100.00	14	4708.44	1/1

10 rows × 25 columns

In [7]:

```
data.tail(10)
```

Out[7]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORI
2813	10293	32	60.06	1	1921.92	9/9/
2814	10306	35	59.51	6	2082.85	1
2815	10315	40	55.69	5	2227.60	1
2816	10327	37	86.74	4	3209.38	1
2817	10337	42	97.16	5	4080.72	1
2818	10350	20	100.00	15	2244.40	
2819	10373	29	100.00	1	3978.51	
2820	10386	43	100.00	4	5417.57	3/1/
2821	10397	34	62.24	1	2116.16	
2822	10414	47	65.52	9	3079.44	5/6/

10 rows × 25 columns

In [9]:

```
data.dtypes
```

Out[9]:

```
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 [10]:

```
data.isnull().sum()
```

Out[10]:

```
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 [11]:

```
data.describe()
```

Out[11]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALE
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.00000
mean	10258.725115	35.092809	83.658544	6.466171	3553.88907
std	92.085478	9.741443	20.174277	4.225841	1841.86510
min	10100.000000	6.000000	26.880000	1.000000	482.13000
25%	10180.000000	27.000000	68.860000	3.000000	2203.43000
50%	10262.000000	35.000000	95.700000	6.000000	3184.80000
75%	10333.500000	43.000000	100.000000	9.000000	4508.00000
max	10425.000000	97.000000	100.000000	18.000000	14082.80000

In [15]:

```
data['ORDERDATE'] = pd.to_datetime(data['ORDERDATE'])
```

In [24]:

```
#We need to create some features in order to create cluseters
#Recency: Number of days between customer's latest order and today's date
#Frequency : Number of purchases by the customers
#Monetary : Revenue generated by the customers

import datetime as dt
snapshot_date = data['ORDERDATE'].max() + dt.timedelta(days = 1)
data_RFM = data.groupby(['CUSTOMERNAME']).agg({
    'ORDERDATE' : lambda x : (snapshot_date - x.max()).days,
    'ORDERNUMBER' : 'count',
    'SALES' : 'sum'
})
```

In [25]:

```
#Rename the columns

data_RFM.rename(columns = {
    'ORDERDATE' : 'Recency',
    'ORDERNUMBER' : 'Frequency',
    'SALES' : 'Monetary'
}, inplace=True)
```

In [26]:

```
data_RFM.head(10)
```

Out[26]:

CUSTOMERNAME	Recency	Frequency	Monetary
AV Stores, Co.	196	51	157807.81
Alpha Cognac	65	20	70488.44
Amica Models & Co.	265	26	94117.26
Anna's Decorations, Ltd	84	46	153996.13
Atelier graphique	188	7	24179.96
Australian Collectables, Ltd	23	23	64591.46
Australian Collectors, Co.	184	55	200995.41
Australian Gift Network, Co	119	15	59469.12
Auto Assoc. & Cie.	233	18	64834.32
Auto Canal Petit	55	27	93170.66

In [28]:

```
# Divide into segments
# We create 4 quartile ranges
data_RFM['M'] = pd.qcut(data_RFM['Monetary'], q = 4, labels = range(1,5))
data_RFM['R'] = pd.qcut(data_RFM['Recency'], q = 4, labels = list(range(4,0,-1)))
data_RFM['F'] = pd.qcut(data_RFM['Frequency'], q = 4, labels = range(1,5))
```

In [30]:

```
#We will now create another column for RFM scores

data_RFM['RFM_Score'] = data_RFM[['R', 'M', 'F']].sum(axis=1)
data_RFM.head()
```

Out[30]:

CUSTOMERNAME	Recency	Frequency	Monetary	M	R	F	RFM_Score
AV Stores, Co.	196	51	157807.81	4	2	4	10
Alpha Cognac	65	20	70488.44	2	4	2	8
Amica Models & Co.	265	26	94117.26	3	1	2	6
Anna's Decorations, Ltd	84	46	153996.13	4	3	4	11
Atelier graphique	188	7	24179.96	1	2	1	4

In [33]:

```
#RFM scores will display the following values:-

#RFM Score > 10 : High Value Customers
#RFM Score < 10 and RFM Score >= 6 : Mid Value Customers
#RFM Score < 6 : Low Value Customers

def rfm_level(data):
    if bool(data['RFM_Score'] >= 10):
        return 'High Value Customer'

    elif bool(data['RFM_Score'] < 10) and bool(data['RFM_Score'] >= 6):
        return 'Mid Value Customer'
    else:
        return 'Low Value Customer'
data_RFM['RFM_Level'] = data_RFM.apply(rfm_level, axis = 1)

#We display the newly created table
data_RFM.head()
```

Out[33]:

CUSTOMERNAME	Recency	Frequency	Monetary	M	R	F	RFM_Score	RFM_Level
AV Stores, Co.	196	51	157807.81	4	2	4	10	High Value Customer
Alpha Cognac	65	20	70488.44	2	4	2	8	Mid Value Customer
Amica Models & Co.	265	26	94117.26	3	1	2	6	Mid Value Customer
Anna's Decorations, Ltd	84	46	153996.13	4	3	4	11	High Value Customer
Atelier graphique	188	7	24179.96	1	2	1	4	Low Value Customer

In [36]:

```
#Now we will be executing KMeans
data = data_RFM[['Recency', 'Frequency', 'Monetary']]
data.head()
```

Out[36]:

CUSTOMERNAME	Recency	Frequency	Monetary
AV Stores, Co.	196	51	157807.81
Alpha Cognac	65	20	70488.44
Amica Models & Co.	265	26	94117.26
Anna's Decorations, Ltd	84	46	153996.13
Atelier graphique	188	7	24179.96

In [37]:

```
# Our data is not accurate so we must remove the unwanted things by performing lo
data_log = np.log(data)
data_log.head()
```

Out[37]:

CUSTOMERNAME	Recency	Frequency	Monetary
AV Stores, Co.	5.278115	3.931826	11.969133
Alpha Cognac	4.174387	2.995732	11.163204
Amica Models & Co.	5.579730	3.258097	11.452297
Anna's Decorations, Ltd	4.430817	3.828641	11.944683
Atelier graphique	5.236442	1.945910	10.093279

In [38]:

```
#Now we will perform standardization
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(data_log)
data_normalized = scaler.transform(data_log)
data_normalized = pd.DataFrame(data_normalized, index = data_log.index, columns=d
data_normalized.describe().round(2)
```

Out[38]:

	Recency	Frequency	Monetary
count	92.00	92.00	92.00
mean	0.00	-0.00	0.00
std	1.01	1.01	1.01
min	-3.51	-3.67	-3.82
25%	-0.24	-0.41	-0.39
50%	0.37	0.06	-0.04
75%	0.53	0.45	0.52
max	1.12	4.03	3.92

In [39]:

```
#Fit KMeans and use elbow method to choose the number of clusters
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans

sse = {}

for k in range(1, 21):
    kmeans = KMeans(n_clusters = k, random_state = 1)
    kmeans.fit(data_normalized)
    sse[k] = kmeans.inertia_
```

In [42]:

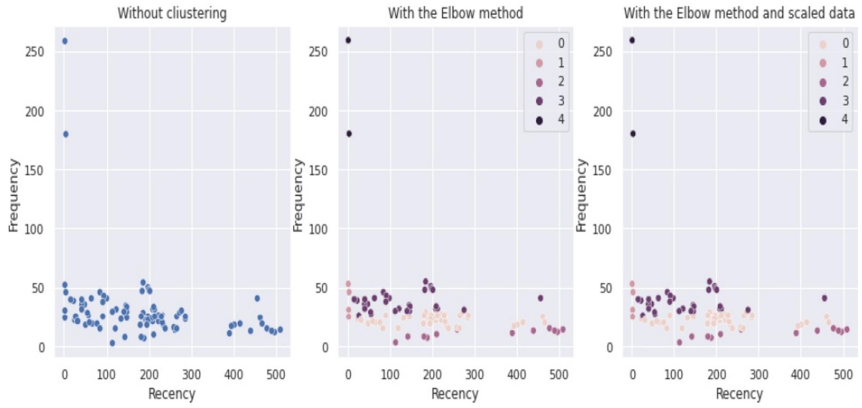
```
#We will make the cluster of size 5
kmeans = KMeans(n_clusters=5, random_state=1)
kmeans.fit(data_normalized)
cluster_labels = kmeans.labels_

data_rfm = data.assign(Cluster = cluster_labels)
data_rfm.head()
```

Out[42]:

CUSTOMERNAME	Recency	Frequency	Monetary	Cluster
AV Stores, Co.	196	51	157807.81	3
Alpha Cognac	65	20	70488.44	0
Amica Models & Co.	265	26	94117.26	0
Anna's Decorations, Ltd	84	46	153996.13	3
Atelier graphique	188	7	24179.96	2

In [55]:

```
from sklearn.cluster import KMeans
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15,5))
sns.scatterplot(ax=axes[0], data=data, x='Recency', y='Frequency').set_title('Wit
sns.scatterplot(ax=axes[1], data=data, x='Recency', y='Frequency', hue=kmeans.lab
sns.scatterplot(ax=axes[2], data=data, x='Recency', y='Frequency', hue=kmeans.lab
```



In [ ]: