Capstone Project Report

Name: **Pooja Ramdas Kadam**
Course: AI & ML (Batch - 4)

Problem Statement

Implement a market customer segmentation using RFM analysis. The necessary steps that you need to perform are:

1. Clean the Data
2. Transform Data for RFM Analysis
3. Perform Clustering on the customer data.

Prerequisites

Along with Python below libraries needed to be installed

Pandas
Numpy
DateTime
Matplotlib
Sklearn
Seaborn

Dataset Used

UIC Online-Retail.xlsx

Implementation

Import required libraries and load data

```
import pandas as pd
import numpy as np
from datetime import timedelta
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

Load data

```
# Read the data set
dfs = pd.read_excel('online-retail.xlsx', sheet_name=None)
```

```
df = dfs['Online Retail']
```

```
df.shape
```

```
(541909, 8)
```

Data Cleaning and Preprocessing

```
df = df.drop_duplicates()
```

```
df.shape
```

```
(536641, 8)
```

```
df = df[(df['Quantity'] > 0) & (df['UnitPrice'] > 0) & (df['CustomerID'].notnull())]
```

```
df.shape
```

```
(392692, 8)
```

:
```
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']
```

:
```
df.head()
```

:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalPrice
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34

Transforming data for RFM Analysis

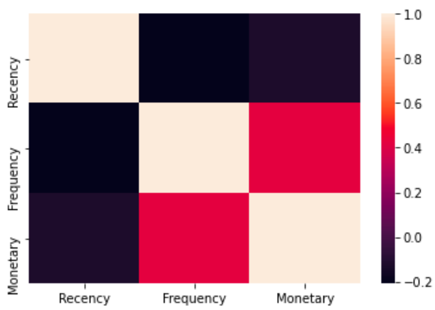
```
rfm_data = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (recent_date - max(x)).days,
    'InvoiceNo': 'count',
    'TotalPrice': 'sum'
})
```

```
rfm_data.corr()
```

	Recency	Frequency	Monetary
Recency	1.000000	-0.206501	-0.121975
Frequency	-0.206501	1.000000	0.425282
Monetary	-0.121975	0.425282	1.000000

```
sns.heatmap(rfm_data.corr())
```

```
<AxesSubplot:>
```



Data Normalization

```
#Normalize the data
scaler = StandardScaler()
```

```
rfm_data = pd.DataFrame(scaler.fit_transform(rfm_data), columns=rfm_data.columns, index=rfm_data.index)
```

```
rfm_data
```

	Recency	Frequency	Monetary
CustomerID			
12346.0	2.329388	-0.397035	8.363010
12347.0	-0.900588	0.405694	0.251699
12348.0	-0.170593	-0.263986	-0.027988
12349.0	-0.740589	-0.077717	-0.032406
12350.0	2.179389	-0.326075	-0.190812
...	...	...	...
18280.0	1.849392	-0.357120	-0.207931
18281.0	0.879399	-0.370425	-0.219037
18282.0	-0.850588	-0.348250	-0.208214
18283.0	-0.890588	2.796139	-0.000352
18287.0	-0.500591	-0.091022	-0.023531

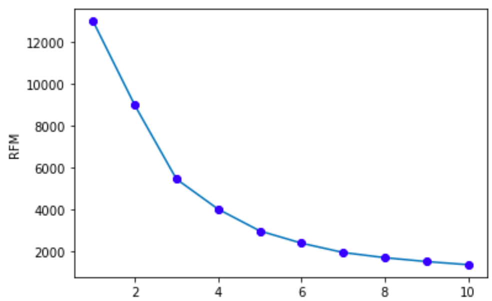
Perform clustering using KMeans

```
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=1).fit(rfm_data)
    inertia.append(kmeans.inertia_)
```

```
inertia
```

```
[13014.000000000004,
 8989.04820478467,
 5446.040054546088,
 4003.948549989731,
 2960.2604240744568,
 2373.0031609399653,
 1925.5803685157377,
 1676.9356946657833,
 1490.0678494944084,
 1341.3142011433988]
```

```
plt.plot(range(1, 11), inertia)
plt.plot(range(1, 11), inertia, 'bo')
plt.xlabel('Number of clusters')
plt.ylabel('RFM')
plt.show()
```




```
#Apply K-Means clustering to cluster the customers
model = KMeans(n_clusters = 5, random_state=1).fit(rfm_data)
centers = model.cluster_centers_
```

```
centers
```

```
array([[-0.76332005,  1.70098836,  1.08000291],
       [-0.48173538, -0.08188082, -0.08013512],
       [ 1.56925152, -0.27992631, -0.17363005],
       [-0.85058848,  3.26107154, 21.0102139 ],
       [-0.90558808, 24.95432826,  7.63157557]])
```

```
rfm_data = pd.DataFrame(scaler.inverse_transform(rfm_data), columns=rfm_data.columns, index=rfm_data.index)
```

```
rfm_data
```

	Recency	Frequency	Monetary
CustomerID			
12346.0	326.0	1.0	77183.60
12347.0	3.0	182.0	4310.00
12348.0	76.0	31.0	1797.24
12349.0	19.0	73.0	1757.55
12350.0	311.0	17.0	334.40
...	...	...	...
18280.0	278.0	10.0	180.60
18281.0	181.0	7.0	80.82
18282.0	8.0	12.0	178.05
18283.0	4.0	721.0	2045.53

Analyse the clusters

```
rfm_data['CustomerID'] = rfm_data.index
rfm_data['Cluster'] = model.labels_
```

```
rfm_data
```

	Recency	Frequency	Monetary	CustomerID	Cluster
CustomerID					
12346.0	326.0	1.0	77183.60	12346.0	0
12347.0	3.0	182.0	4310.00	12347.0	1
12348.0	76.0	31.0	1797.24	12348.0	1
12349.0	19.0	73.0	1757.55	12349.0	1
12350.0	311.0	17.0	334.40	12350.0	2
...	...	...	...	...	...
18280.0	278.0	10.0	180.60	18280.0	2
18281.0	181.0	7.0	80.82	18281.0	2
18282.0	8.0	12.0	178.05	18282.0	1
18283.0	4.0	721.0	2045.53	18283.0	0
18287.0	43.0	70.0	1837.28	18287.0	1

4338 rows × 5 columns

```
rfm_data.groupby('Cluster').agg({
    'Recency': ['min', 'mean', 'max'],
    'Frequency': ['min', 'mean', 'max'],
    'Monetary': ['min', 'mean', 'max'],
})
```

	Recency			Frequency			Monetary		
	min	mean	max	min	mean	max	min	mean	max
Cluster									
0	1.0	16.726908	326.0	1.0	474.064257	2677.0	1071.73	11751.644297	91062.38
1	1.0	44.885582	157.0	1.0	72.061177	342.0	6.20	1328.738576	16209.50
2	146.0	249.985782	374.0	1.0	27.405687	297.0	3.75	488.761897	9864.26
3	1.0	8.000000	25.0	3.0	825.833333	2076.0	117210.08	190808.536667	280206.02
4	1.0	2.500000	5.0	4412.0	5717.250000	7676.0	33053.19	70612.247500	143711.17

Conclusions

Cluster 4 has most valuable customers where they spend more with recent purcharchase history, so they can be targeted for new product launches ¶

Cluster 2 has the most probably lost customers, need to make survey and get feedback from them and improve the services

Cluster 3 has more Monetary value, and avg frequency. Can target them with ads to get them to buy more and more