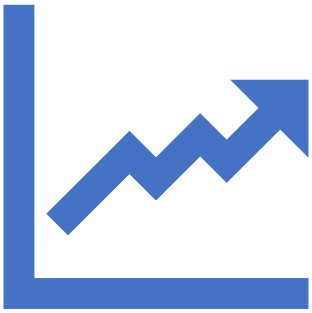

RFM For Sales Analysis

-Chetna Soni

What is RFM Model?

RFM segmentation is a great method to identify groups of customers for special treatment. This method to improve customer marketing.

RFM analysis allows marketers to target specific clusters of customers with communications that are much more relevant for their particular behavior – and thus generate much higher rates of response, plus increased loyalty and customer lifetime value . Like other segmentation methods, an RFM model is a powerful way to identify groups of customers for special treatment. RFM stands for **R**ecency, **F**requency and **M**onetary.

What are Recency, Frequency and Monetary?

- Underlying the RFM segmentation technique is the idea that marketers can gain an extensive understanding of their customers by analyzing three quantifiable factors. These are:
- **Recency**: How much time has elapsed since a customer's last activity or transaction with the brand? Activity is usually a purchase, although variations are sometimes used, e.g., the last visit to a website or use of a mobile app. In most cases, the more recently a customer has interacted or transacted with a brand, the more likely that customer will be responsive to communications from the brand.
- **Frequency**: How often has a customer transacted or interacted with the brand during a particular period of time? Clearly, customers with frequent activities are more engaged, and probably more loyal, than customers who rarely do so. And one -time-only customers are in a class of their own.
- **Monetary**: Also referred to as "monetary value," this factor reflects how much a customer has spent with the brand during a particular period of time. Big spenders should usually be treated differently than customers who spend little. Looking at monetary divided by frequency indicates the average purchase amount – an important secondary factor to consider when segmenting customers.

Important key points in dataset.

- **Best Customers** – Communications with this group should make them feel valued and appreciated. These customers likely generate a disproportionately high percentage of overall revenues and thus focusing on keeping them happy should be a top priority. Further analyzing their individual preferences and affinities will provide additional opportunities for even more personalized messaging.
- **High-spending New Customers** – It is always a good idea to carefully "incubate" all new customers, but because these new customers spent a lot on their first purchase, it's even more important. Like with the Best Customers group, it's important to make them feel valued and appreciated – and to give them terrific incentives to continue interacting with the brand.
- **Lowest-Spending Active Loyal Customers** – These repeat customers are active and loyal, but they are low spenders. Marketers should create campaigns for this group that make them feel valued, and incentivize them to increase their spend levels. As loyal customers, it often also pays to reward them with special offers if they spread the word about the brand to their friends, e.g., via social networks.
- **Churned Best Customers** – These are valuable customers who stopped transacting a long time ago. While it's often challenging to re -engage churned customers, the high value of these customers makes it worthwhile trying. Like with the Best Customers group, it's important to communicate with them on the basis of their specific preferences, as known from earlier transaction data.

Model/Method used to Identify potential Customer

- **K-Means Clustering -** It is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.
- K-Mean Clustering Model - Cluster analysis uses mathematical models to discover groups of similar customers based on the smallest variations among customers within each group.
- **Customer Segmentation -** Customer segmentation analysis is the process performed when looking to discover insights that define specific segments of customers. Marketers and brands leverage this process to determine what campaigns, offers, or products to leverage when communicating with specific segments.
- Customer segmentation has the potential to allow marketers to address each customer in the most effective way. Using the large amount of data available on customers (and potential customers), a customer segmentation analysis allows marketers to identify discrete groups of customers with a high degree of accuracy based on demographic, behavioral and other indicators.

EDA (Exploratory Data Analysis) using RFM model

- Import libraries and load dataset

```
import numpy as np
 import pandas as pd
import matplotlib.pyplotas plt
%matplotlib inline
import seaborn as sns
```

```
import numpy as np import pandas as pd importmatplotlib.pyplotas plt %matplotlib
inline import seaborn as sns
```

- Check for missing values - No Missing value present in this dataset.
- Finding correlation between features given in dataset.
- Plotting the data on the basis of “Revenue”, Dayssincelastorder“, "Total_orders “. Using Matplotlib Library.

Summary From EDA and Modelling

- **Champions** – This group consists of those customers who are found in R -Tier-1, F-Tier-1 and M-Tier-1, meaning that they transacted recently, do so often and spend more than other customers. A shortened notation for this segment is 1 -1-1; we'll use this notation going forward.
- **Potential customers** – This group consists of those customers in 1 -3-1 and 1-3-2. These are customers who transacted only once, but very recently and they spent a lot.
- **Need attention**– This group consists of those customers in segments 1 -1-2 and 1-1-3 (they transacted recently and do so often, but spend the least).

CODE ::

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# In[4]:


df= pd.read_excel(r"C:\Users\chetn\Desktop\chetnapython\Round 1 Assignment\sales_data.xlsx")


# In[5]:


df.head()


# In[6]:
```

```
rmf_df = pd.DataFrame({
    "customer" : df["CustomerID"],
    "recency" : df["DAYSSINCELASTORDER"],
    "monetary" : df["REVENUE"],
    "frequency" : df["TOTAL_ORDERS"]
})


# In[155]:


rmf_df


# In[9]:


rmf_df.shape[0]


# In[7]:


rmf_df = rmf_df.sort_values(["recency"],ascending=True)
divid = int(rmf_df.shape[0]/3)

li = [1] * rmf_df.shape[0]
li[:divid] = [3] * divid
li[divid:divid*2] = [2] * divid
```

```
rmf_df["recency_rank"] = li



display(rmf_df)
# ________

rmf_df = rmf_df.sort_values(["frequency"],ascending=False)
divid = int(rmf_df.shape[0]/3)

li = [1] * rmf_df.shape[0]
li[:divid] = [3] * divid
li[divid:divid*2] = [2] * divid

rmf_df["frequency_rank"] = li
display(rmf_df)

# ________

rmf_df = rmf_df.sort_values(["monetary"],ascending=False)
divid = int(rmf_df.shape[0]/3)

li = [1] * rmf_df.shape[0]
li[:divid] = [3] * divid
li[divid:divid*2] = [2] * divid

rmf_df["monetary_rank"] = li



# In[ ]:
```

In[157]:

rmf_df

In[161]:

RMF score calculation

rmf_df["score"] = (0.15 * rmf_df["recency_rank"]) + (0.28 * rmf_df["frequency_rank"]) + (0.57 * rmf_df["monetary_rank"])

In[ ]:

Score

rmf_df["score"]

In[162]:

rmf_df["clusters"]=1

In[163]:

```
rmf_df.loc[list(rmf_df["score"].sort_values()[:1667].index), 'clusters']=3
rmf_df.loc[list(rmf_df["score"].sort_values()[1667:1667*2].index), 'clusters']=2
```

In[164]:

```
rmf_df
```

In[165]:

```
cluster_1 = rmf_df.iloc[list(rmf_df["score"].sort_values()[:1667].index)]
```

In[166]:

```
cluster_2 = rmf_df.iloc[list(rmf_df["score"].sort_values()[1667:1667*2].index)]
```

In[167]:

```
cluster_3 = rmf_df.iloc[list(rmf_df["score"].sort_values()[1667*2:].index)]


# In[168]:


import numpy as np

from scipy.stats import norm
import statistics

# Plot between -10 and 10 with .001 steps.
x_axis = rmf_df["monetary"]

# Calculating mean and standard deviation
mean = statistics.mean(x_axis)
sd = statistics.stdev(x_axis)
print(mean,sd)
plt.scatter(x_axis, norm.pdf(x_axis, mean, sd),c=rmf_df["clusters"])

plt.show()


# In[169]:


# plt.scatter(old_rmf_df.index,old_rmf_df["recency"],c=rmf_df["clusters"])
plt.scatter(old_rmf_df.index,old_rmf_df["monetary"],c=rmf_df["clusters"])
# plt.scatter(old_rmf_df.index,old_rmf_df["frequency"],c=rmf_df["clusters"])
```

```
# In[170]:

old_rmf_df = rmf_df.sort_index()

# In[ ]:

# In[142]:

old_rmf_df[old_rmf_df["clusters"]==2]

# In[143]:

3032/21

# In[12]:

#Using K-mean Model for Analysis
```

```
pip install scikit-learn

# In[13]:

from sklearn.cluster import KMeans

# In[14]:

kmeans = KMeans(n_clusters=3,
random_state=0).fit(df[["REVENUE","TOTAL_ORDERS","DAYSSINCELASTORDER"]])

# In[15]:

df.describe()

# In[16]:

kmeans.predict(df[["REVENUE","TOTAL_ORDERS","DAYSSINCELASTORDER"]])

# In[17]:
```

```
df['kmeans_labels']= kmeans.predict(df[["REVENUE","TOTAL_ORDERS","DAYSSINCELASTORDER"]])


# In[18]:


df


# In[19]:


pip install matplotlib


# In[21]:


df = df.drop('CustomerID',1)


# In[22]:


df


# In[23]:
```

```
import matplotlib.pyplot as plt
for col in df.columns:
    plt.figure(figsize=(10,5))
    plt.scatter(df.index,df[col],c=df["kmeans_labels"])
    plt.title(col)
    plt.show()


# In[ ]:
```