Customer Segmentation for E-Commerce Businesses using K-means Algorithm

Presented to
DR. MOHAMMAD ASHRAFUZZAMAN KHAN

Presented by:
HM Adnan Zami 2115164650

E-Commerce Industry

- “According to the latest data from Statista, the e-commerce market in Bangladesh in 2019 stood at 1,648 million USD which will increase to 2,077 million USD this year and in 2023, the market size will be 3,077 million USD.” [New Age BD, 2020]

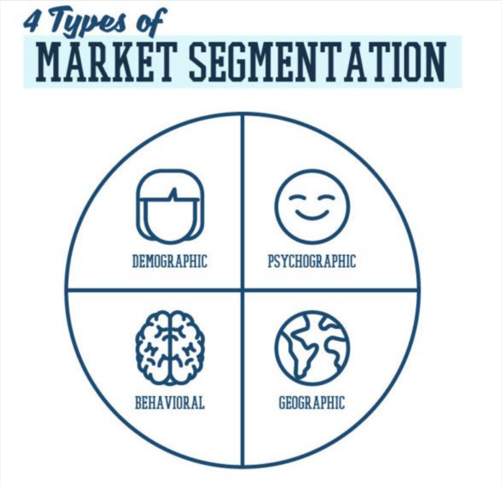
- Challenges
    - Attracting the perfect customer
    - Generating targeted traffic
    - Capturing quality leads
    - Retaining customers

Problem Statement

- Growth of e-commerce business has increased the competition making it difficult for businesses to improve customer acquisition and retention.

Solution

Customer Segmentation - process of dividing customers into groups based on common characteristics so companies can execute targeted marketing to each group effectively and appropriately.



1. Demographic segmentation -
   a. Age,
   b. Gender,
   c. Ethnicity
2. Psychographic segmentation -
   a. Personality traits,
   b. Hobbies,
   c. Beliefs,
   d. Lifestyles
3. Behavioral segmentation -
   a. Spending habits,
   b. Purchasing habits
   c. Browsing habits
4. Geographic segmentation -
   a. Country,
   b. Region,
   c. City

Methodology

Objective - identify customer habits to develop targeted marketing to increase conversion rates for e-commerce businesses. (Behavioral Segmentation & Geographical)

RFM Analysis & Country -

- Recency - days since last purchase
- Frequency - total number of invoices
- Monetary value - total amount spent by a customer.
- Country

Looking for patterns in a non-labeled dataset for which unsupervised machine learning needs to be implemented. We chose k-means to classify the dataset.

Dataset & Feature Selection

Dataset - This Online Retail II data set contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique all-occasion giftware.

Total - 1,067,371 rows

Attributes - 8

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

df.head()

Dataset & Feature Selection

Dataset - This Online Retail II data set contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique all-occasion giftware.

- Selected a total of 541,910 rows starting from 1/1/2010 - 12/09/2011.
- After cleaning the rows with NULL values using dropna(), we remained with **397,925 rows** with **4339 customers**

Selected Features for RFM Analysis -

No.	Attributes	Feature Analysis
1	Invoice Date	Recency
2	Invoices	Frequency
3	Unit Price	Monetary Value
4	Purchase Quantity	
5.	Country	Country

RFM Calculation

- Recency - since data is between 2010 and 2011, the date used to compute the Number of Days Since Last Purchase was 1/1/2012.

```
#Recency

rec = []  #empty list to store number of days since last purchase

date_from = ['1/1/2012  00:00']   #present date, compared with purchased date
date_from = pd.to_datetime(date_from)

#loop unique customers
for x in u_customer:
    row_customer = df.loc[df['Customer ID'] == x]  #select rows with Customer ID
    inv_dates = row_customer.InvoiceDate  #store invoice date
    recent_date = max(inv_dates)   #find latest purchase date
    recent_date = pd.to_datetime(recent_date)
    num_days = date_from - recent_date    #calculate number of days
    conv_days = num_days.days[0]
    rec.append(conv_days)       #store number of days

recency = {
    'Customer ID': u_customer,
    'Days Since Last Purchase': rec
}


r_analysis = pd.DataFrame(recency, columns = ['Customer ID', 'Days Since Last Purchase'])
r_analysis
```

	Customer ID	Days Since Last Purchase
0	12346.0	347
1	12347.0	151
2	12348.0	97
3	12349.0	40
4	12350.0	332
...	...	...
4334	18280.0	299
4335	18281.0	202
4336	18282.0	148
4337	18283.0	117
4338	18287.0	223

4339 rows × 2 columns

RFM Calculation

- Monetary value - total money spent (Price * Quantity)

```
#Monetary

monetary_value = [] #empty list to store total money spent

for x in u_customer:   #iterate through unique customers
    row_customer = df.loc[df['Customer ID'] == x]  #iterate unique customers
    unit_price = row_customer.Price  #store price for unique customer
    quantity = row_customer.Quantity #store quantity for unique customer
    invoice_money = unit_price * quantity #calculate money spent for each row
    total_money = invoice_money.sum() #sum total money spent
    monetary_value.append(total_money) #store total money spent


cus_mon = {
    'Customer ID': u_customer,
    'Total Money Spent': monetary_value,
}

m_analysis = pd.DataFrame(cus_mon, columns = ['Customer ID', 'Total Money Spent'])

m_analysis
```

	Customer ID	Total Money Spent
0	12346.0	77183.60
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40
...	...	...
4334	18280.0	180.60
4335	18281.0	80.82
4336	18282.0	178.05
4337	18283.0	2094.88
4338	18287.0	1837.28

4339 rows × 2 columns

RFM Calculation

- Frequency - total number of invoices

```python
# Frequency

items = [] #empty list to store all invoices of each  customer
invoices = [] #empty list to store all unique invoices of each customer

for x in u_customer: #iterate through unique customers
    row_customer = df.loc[df['Customer ID'] == x]
    inv = row_customer.Invoice #select all invoices for each customer
    s_inv = set(inv)  #select unique invoices each customer
    f_inv = inv.count() #count total invoices for each customer
    u_inv = len(s_inv) #count total unique invoices
    items.append(f_inv) #add invoices to empty item list
    invoices.append(u_inv) #add invoices to empty invoice list


inv_record = {
    'Customer ID': u_customer,
    'Items': items,
    'Invoices': invoices
}

f_analysis = pd.DataFrame(inv_record, columns = ['Customer ID', 'Items', 'Invoices'])

f_analysis
```

	Customer ID	Items	Invoices
0	12346.0	1	1
1	12347.0	182	7
2	12348.0	31	4
3	12349.0	73	1
4	12350.0	17	1
...	...	...	...
4334	18280.0	10	1
4335	18281.0	7	1
4336	18282.0	12	2
4337	18283.0	756	16
4338	18287.0	70	3

4339 rows × 3 columns

RFM Calculation

	Customer ID	Items	Invoices	Total Money Spent	Days Since Last Purchase
0	12346.0	1	1	77183.60	347
1	12347.0	182	7	4310.00	151
2	12348.0	31	4	1797.24	97
3	12349.0	73	1	1757.55	40
4	12350.0	17	1	334.40	332
...	...	...	...	...	...
4334	18280.0	10	1	180.60	299
4335	18281.0	7	1	80.82	202
4336	18282.0	12	2	178.05	148
4337	18283.0	756	16	2094.88	117
4338	18287.0	70	3	1837.28	223

4339 rows × 5 columns

Setting RFM Ranks

- Ranks were given from 1-5 with 5 being the highest.
- RFM results were broken in percentiles and then assigned using the following table.

Ranks	Recency (R)	Frequency (F)	Monetary Value (M)
1	(0.8) =< R1	F < (0.2)	M < (0.2)
2	(0.6) <= R < (0.8)	(0.2) <= F < (0.4)	(0.2) <= M < (0.4)
3	(0.4) <= R < (0.6)	(0.4) <= F < (0.6)	(0.4) <= M < (0,6)
4	(0.2) <= R < (0.4)	(0.6) <= F < (0.8)	(0.6) <= M < (0.8)
5	R < (0.2)	(0.8) =< F	(0.8) =< M

- Used quantile() function to find the percentile.

	Customer ID	Items	Invoices	Total Money Spent	Days Since Last Purchase	Recency Rank	Frequency Rank	Monetary Rank	RFM Score
0.2	13505.6	14.0	1.0	250.106	93.0	1.0	2.0	1.6	6.0
0.4	14708.2	29.0	2.0	489.724	114.0	2.0	3.0	2.2	9.0
0.6	15882.8	58.0	3.0	941.942	158.0	3.0	4.0	3.8	10.0
0.8	17080.4	121.0	6.0	2057.914	240.0	4.0	5.0	4.4	12.0

Setting RFM Ranks

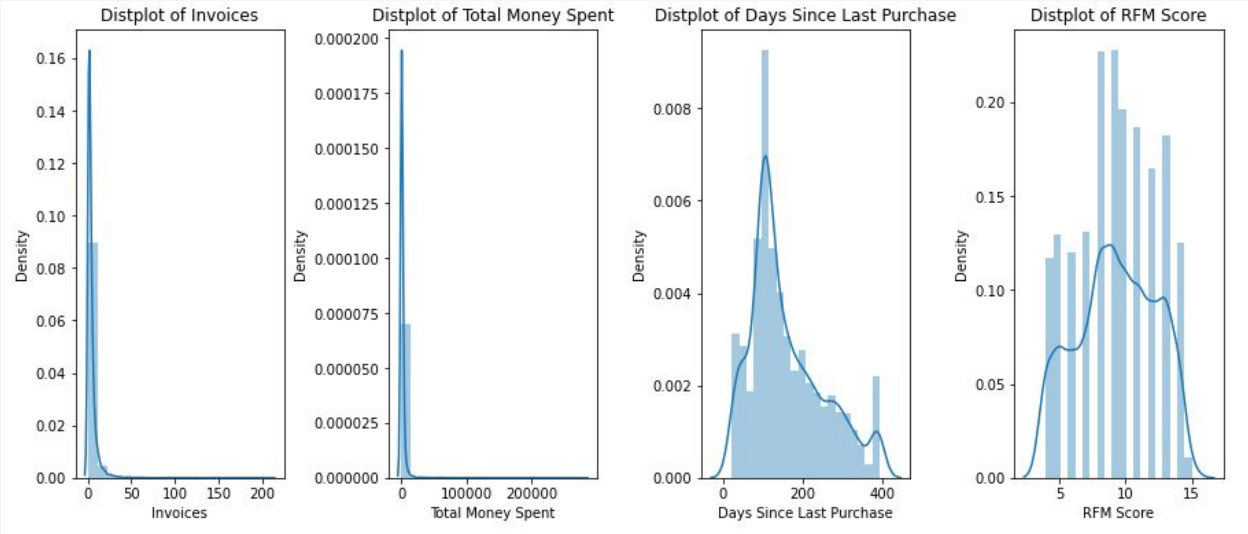
- Computed the **RFM score** by summing R, F and M ranks.

	Customer ID	Items	Invoices	Total Money Spent	Days Since Last Purchase	Recency Rank	Frequency Rank	Monetary Rank	RFM Score	Country
0	12346.0	1	1	77183.60	347	1	2	5	8	United Kingdom
1	12347.0	182	7	4310.00	151	3	5	5	13	Iceland
2	12348.0	31	4	1797.24	97	4	4	4	12	Finland
3	12349.0	73	1	1757.55	40	5	2	4	11	Italy
4	12350.0	17	1	334.40	332	1	2	2	5	Norway
...	...	...	...	...	...	...	...	...	...	...
4334	18280.0	10	1	180.60	299	1	2	1	4	United Kingdom
4335	18281.0	7	1	80.82	202	2	2	1	5	United Kingdom
4336	18282.0	12	2	178.05	148	3	3	1	7	United Kingdom
4337	18283.0	756	16	2094.88	117	3	5	5	13	United Kingdom
4338	18287.0	70	3	1837.28	223	2	4	4	10	United Kingdom

4339 rows × 10 columns

Distance Plot

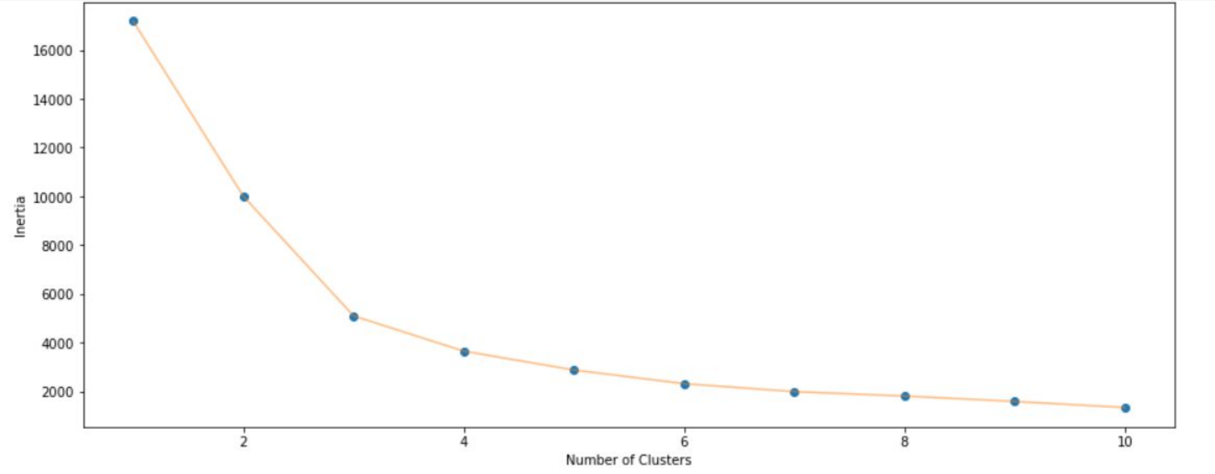
- Shows the distribution of each rank (R, F & M) and RFM score.
- RFM score shows majority of the customers fall in the 8-10 RFM score
- Most customers purchased products around 100 days ago from 1/1/2012.



Segmentation with K-means clustering

Recency & Monetary

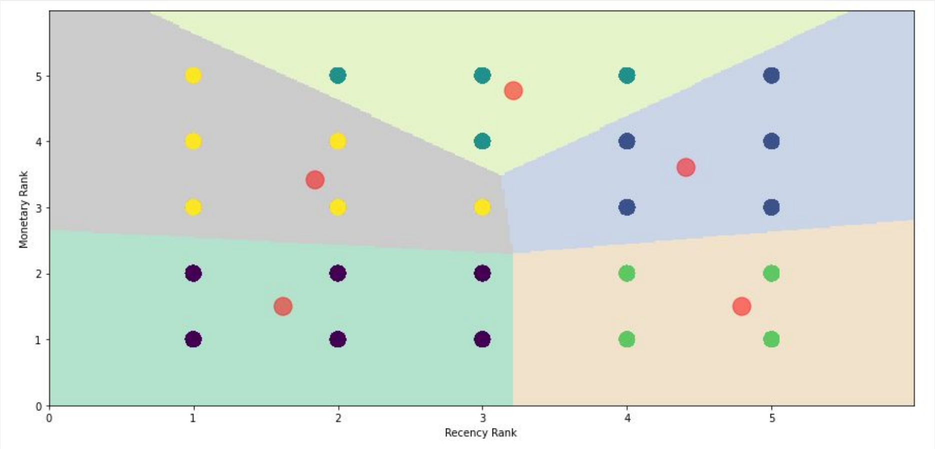
1. Find the value of ‘k’ using Elbow method
    a. Calculate the WCSS (within cluster sum of squares) which is the inertia $$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$
    b. Select the point with lowest change between the next and previous value as the value for ‘k’



Segmentation with K-means clustering

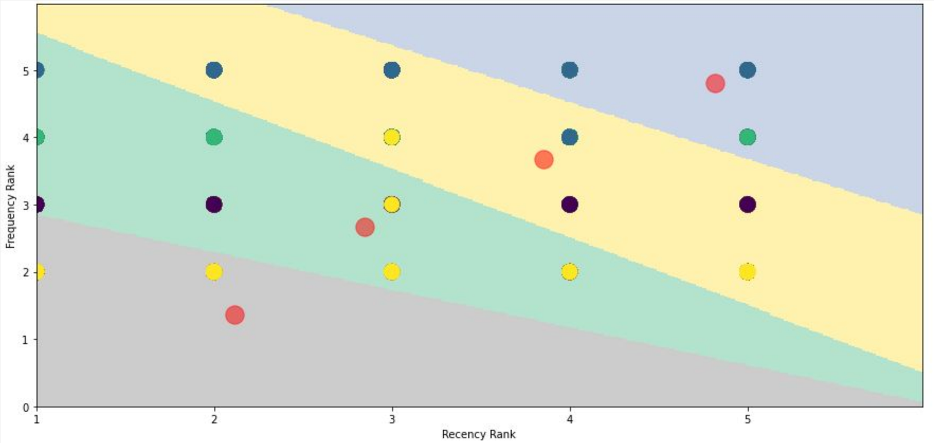
Recency Rank & Monetary Rank

2. Perform clustering for 5 clusters (k=5)
    a. Select random centeroids
        i. Define boundaries by finding the average Euclidean distance between other centeroids
    b. Find Euclidean distance of data points within cluster and average the values
    c. Adjust centroid to the averaged value and iterate step a (performed 300 iterations)

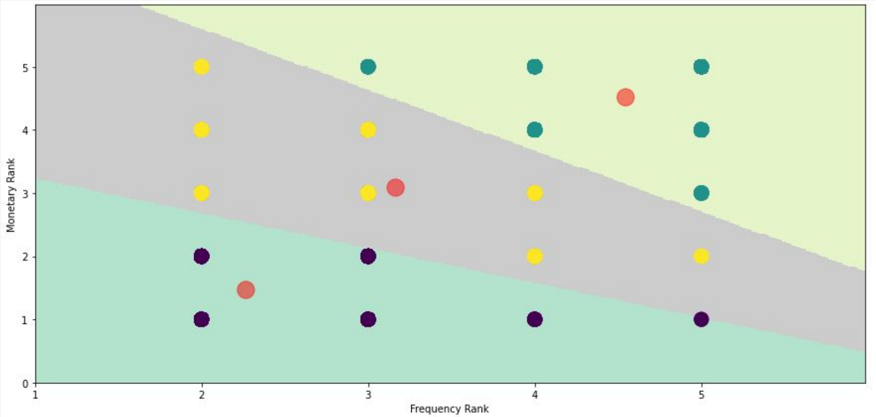


Segmentation with K-means clustering

Recency Rank & Frequency Rank

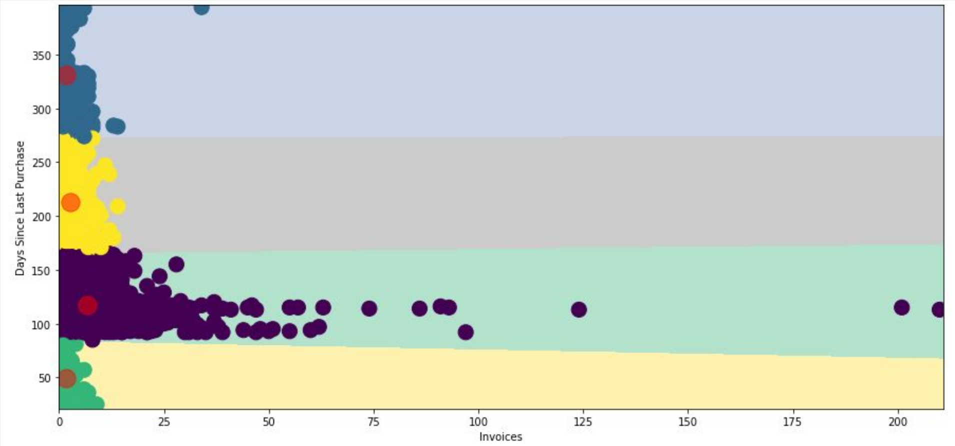


Frequency Rank & Monetary Rank

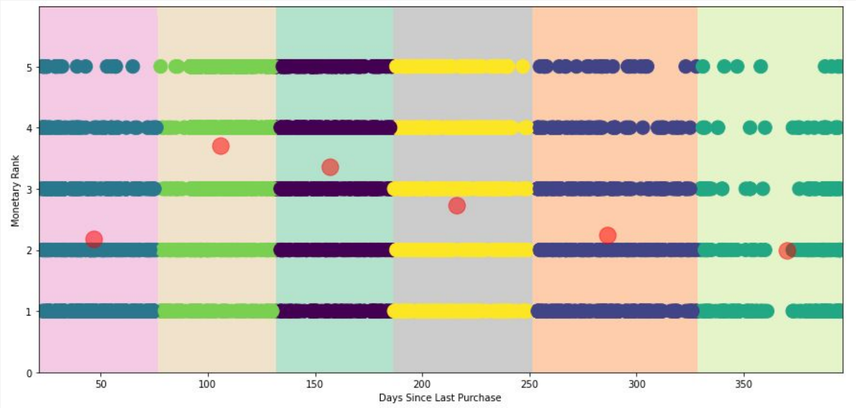


Segmentation with K-means clustering

Invoices & Days Since Last Purchase



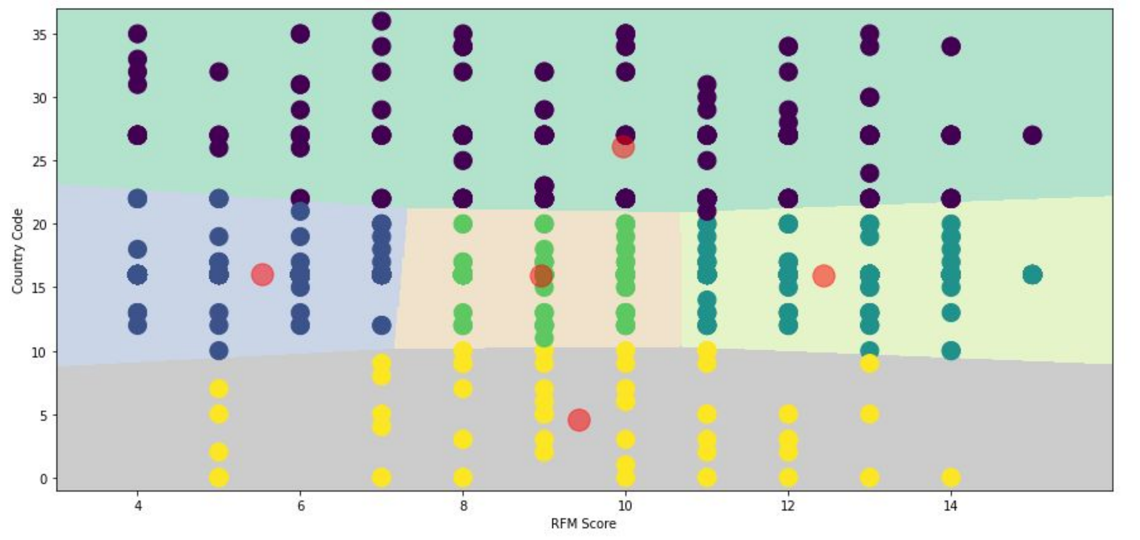
Days Since Last Purchase & Monetary Rank



Segmentation with K-means clustering

RFM Score & Country

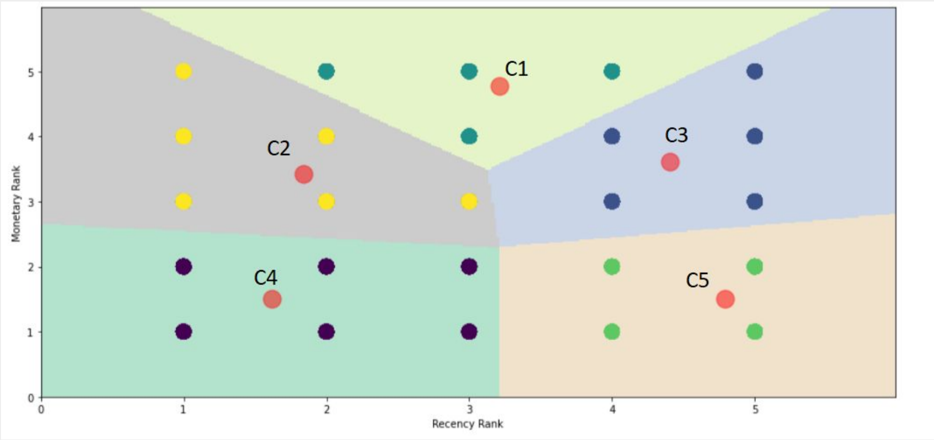
	Country	Country Code
0	Australia	0
1	Austria	1
2	Bahrain	2
3	Belgium	3
4	Brazil	4
5	Canada	5
6	Channel Islands	6
7	Cyprus	7
8	Czech Republic	8
9	Denmark	9
10	EIRE	10
11	European Community	11
12	Finland	12
13	France	13
14	Germany	14
15	Greece	15
16	Iceland	16
17	Israel	17
18	Italy	18
19	Japan	19
20	Lebanon	20
21	Lithuania	21
22	Malta	22
23	Netherlands	23
24	Norway	24
25	Poland	25
26	Portugal	26
27	RSA	27
28	Saudi Arabia	28
29	Singapore	29
30	Spain	30
31	Sweden	31
32	Switzerland	32
33	USA	33
34	United Arab Emirates	34
35	United Kingdom	35
36	Unspecified	36



Analysis

Recency Rank (x-axis) & Monetary Rank (y-axis)

1. C1- high spending customers with average activity
2. C2 - high spending customer but not active
3. C3 - highly frequent customer with high spending
4. C4 - low spending customer who are not active
5. C5 - highly active customer with low spending

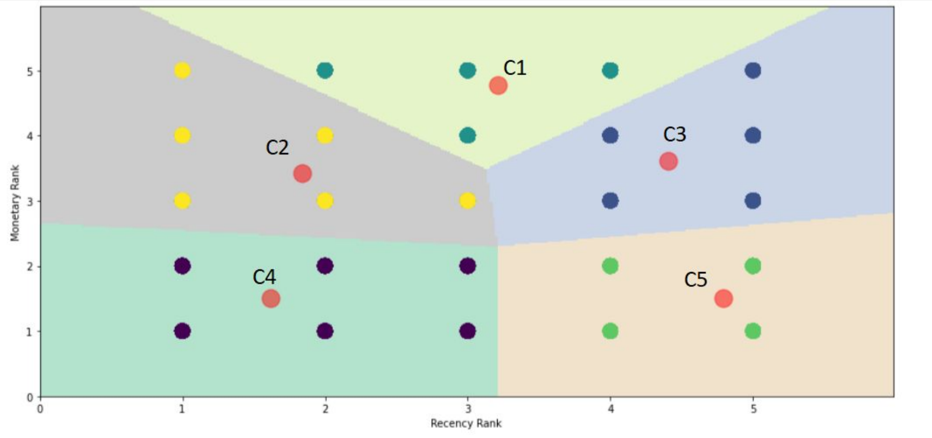


Analysis

Targeted Marketing Strategies

1. C1- Advertise high priced products once in a while as their activity is average but spending rank is high
2. C2 - Must create marketing campaigns to convert them into loyal customers as these users have a tendency to spend high but are not recent.
3. C3 - Best customer group. Always include them when marketing high priced products. Provide loyalty benefits
4. C4 - Customers at risk. Provide discounts or special offers to increase engagement.
5. C5 - Provide discounts or offers as these customers are highly active. Can perform A/B testing for marketings campaigns of new products using these customers

Recency Rank (x-axis) & Monetary Rank (y-axis)

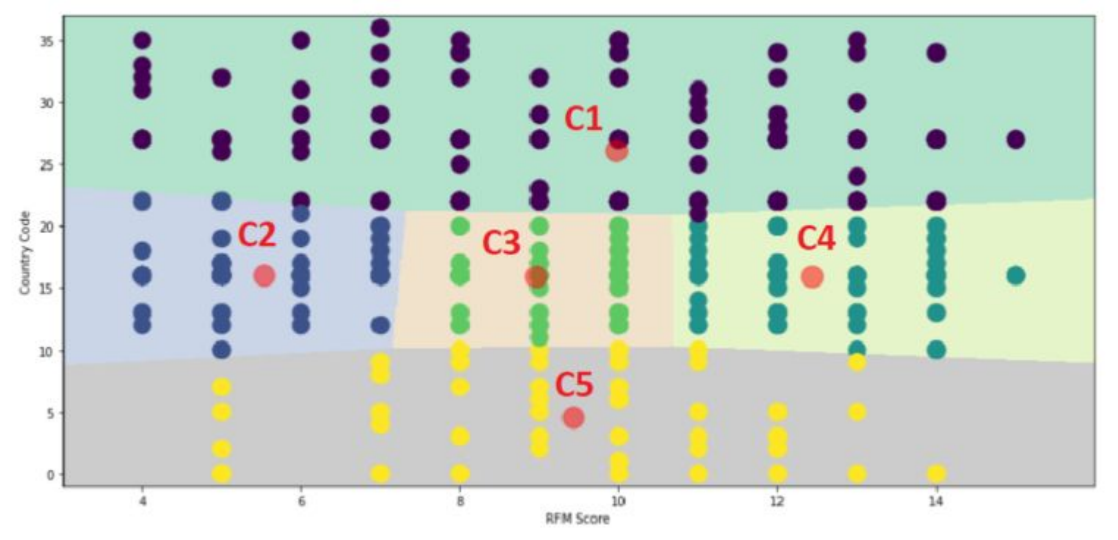


Analysis

RFM Score & Country

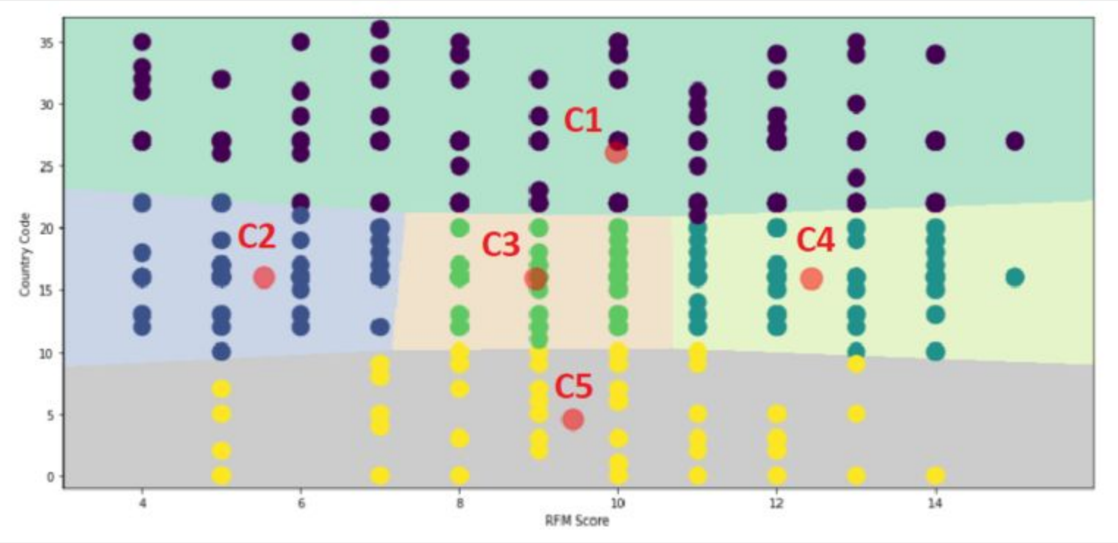
	Country	Country Code
0	Australia	0
1	Austria	1
2	Bahrain	2
3	Belgium	3
4	Brazil	4
5	Canada	5
6	Channel Islands	6
7	Cyprus	7
8	Czech Republic	8
9	Denmark	9
10	EIRE	10
11	European Community	11
12	Finland	12
13	France	13
14	Germany	14
15	Greece	15
16	Iceland	16
17	Israel	17
18	Italy	18
19	Japan	19
20	Lebanon	20
21	Lithuania	21
22	Malta	22
23	Netherlands	23
24	Norway	24
25	Poland	25
26	Portugal	26
27	RSA	27
28	Saudi Arabia	28
29	Singapore	29
30	Spain	30
31	Sweden	31
32	Switzerland	32
33	USA	33
34	United Arab Emirates	34
35	United Kingdom	35
36	Unspecified	36

RFM Score (x-axis) & Country Code (y-axis)



Analysis

RFM Score (x-axis) & Country Code (y-axis)



RFM Score & Country

C1 - Covers all RFM score range. This data is not helpful as we cannot differentiate whether the customers are in the higher RFM range or lower.

C2 - Needs attention. Try to create marketing campaigns targeted to gain these customers interest for your businesses.

C3 - Regularly send marketing content to these customers and try to improve their activity in your business. These customers can be converted into high value customers.

C4 - Best customer segment. Provide loyalty bonuses to maintain these customers in your business

C5 - Similar to C1, this cluster does not provide enough information as customers from these countries fall in all RFM score range.

Conclusion

Customer segmentation is a method of improving customer relations by learning about the customer's wants and activities so that appropriate market strategies can be established. In this project, we were successful at performing the following:

- Find patterns in a non-labeled dataset, unsupervised machine learning
- Identifying ranks for customers to compute the RFM Score
- Applying K-means algorithm to perform customer segmentation
- Classify the customers into their clusters
- Preparing targeted marketing strategies for each cluster of customers.