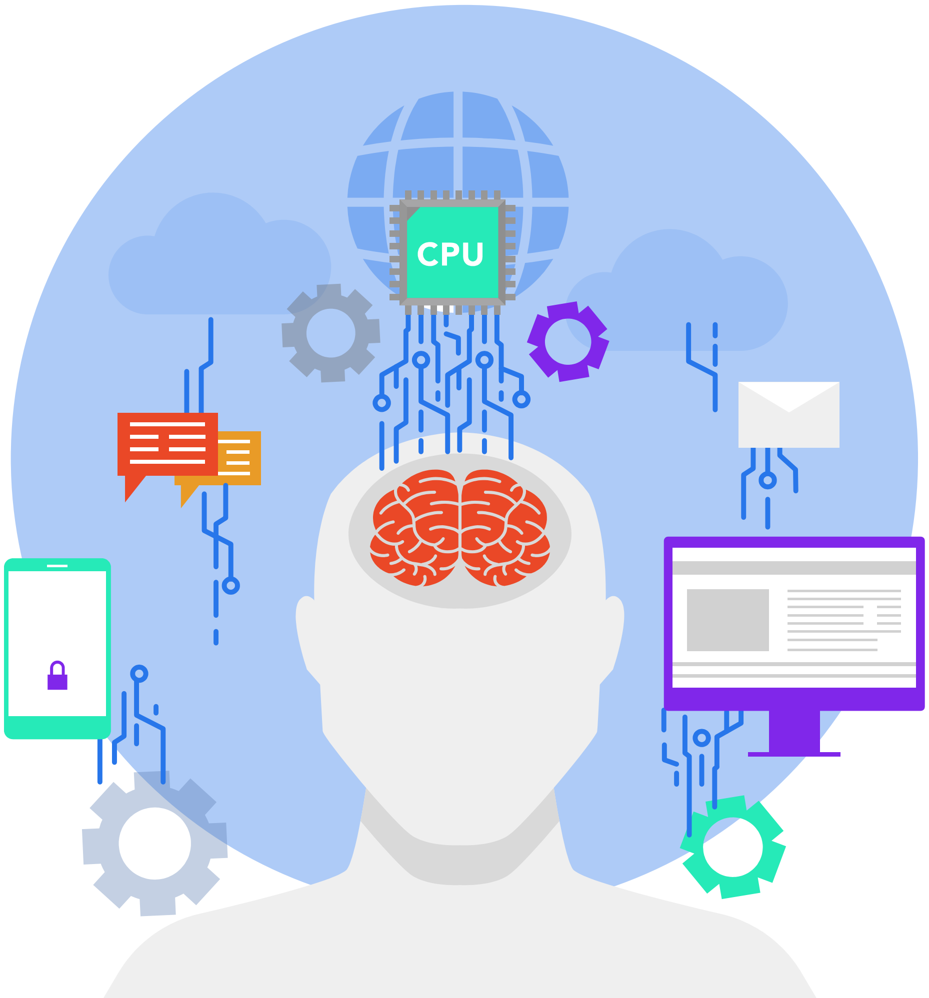


Customer Segmentation RFM

Fauzia Y. Ayupuspita

Background & Goals

Background

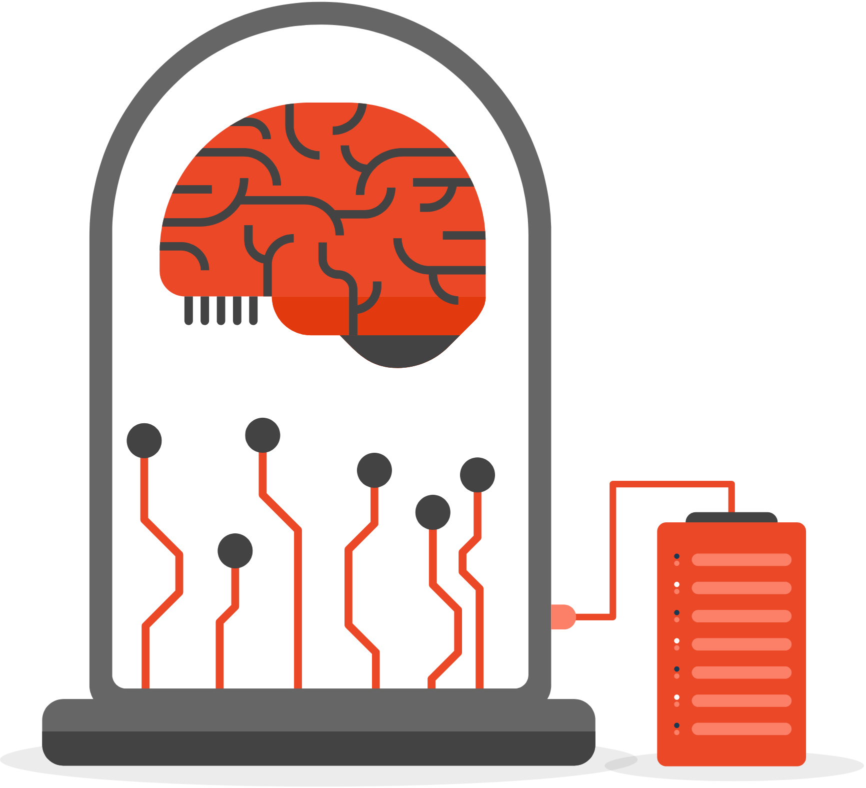
Rachel is the owner e-commerce start up based in Bangkok. Unfortunately, Rachel is launching her product during Covid-19 hits and making her business grow slower than ever. Besides, she hasn’t use targeted marketing which hurt her marketing budget.

Goals

Help to increase Rachel’s marketing conversation rate by doing more targeted market using **customer segmentation**, so that **will not hurt her budget**.

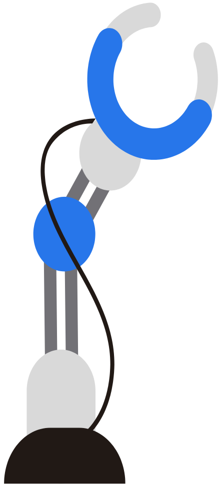
Disclamer !

The dataset that used for this report **has been cleaned**, so the process of merging, removing unnecessary columns, outliers, typos, duplicate rows, and null has been done prior to this part.



Check Dataset Info

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 397884 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    397884 non-null  object
 1   StockCode    397884 non-null  object
 2   Description  397884 non-null  object
 3   Quantity     397884 non-null  int64
 4   InvoiceDate  397884 non-null  datetime64[ns]
 5   UnitPrice    397884 non-null  float64
 6   CustomerID   397884 non-null  float64
 7   Country      397884 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 27.3+ MB
```



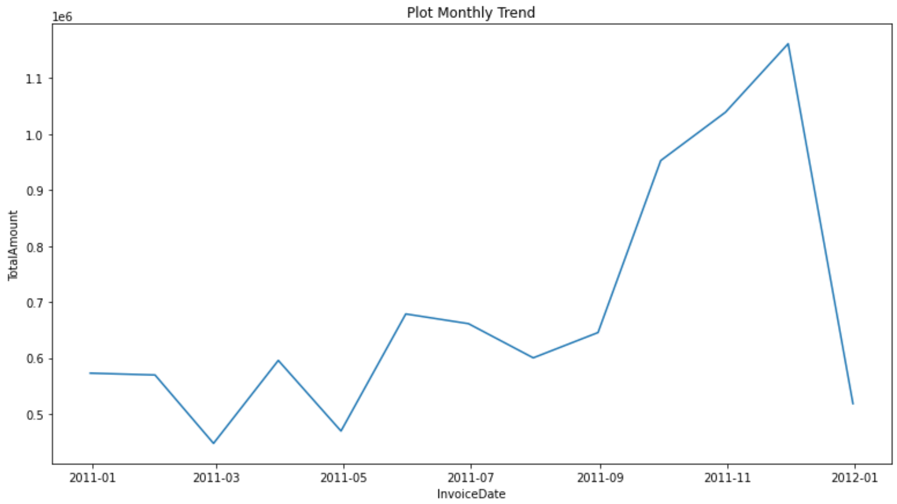
I will use **Recency, Frequency, and Monetary Analysis (RFM).**

I created 2 columns extra to define better performance of my cluster segmentation analysis. The 2 columns are:

1. Total Amount
2. Country is UK

Creating Total Amount

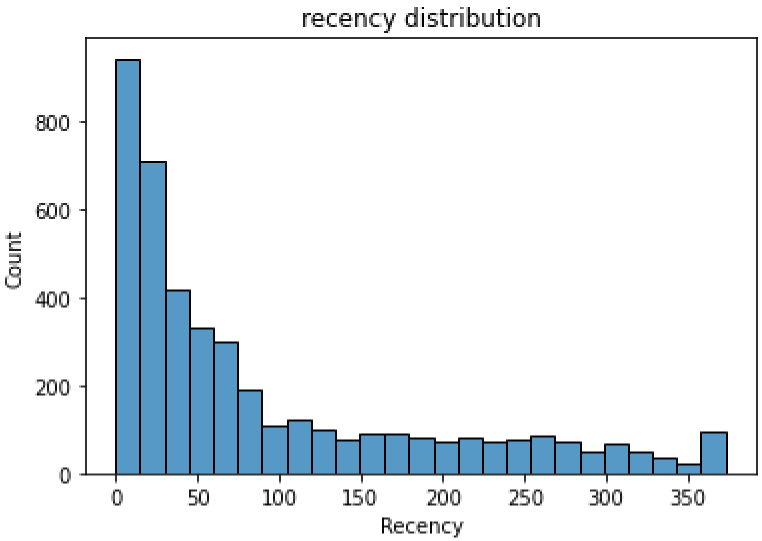
The purpose making this column is to provide an alternative perspective to ‘Invoice Date’ as the base of analysis.



As we can see, **the highest** Total Amount in range period January 2011- January 2012 is in **December 2012**. It can be, because in the end of the year, customer get a lot of money to spend their holiday.

Recency

```
recency = df.groupby(by=['CustomerID'])['InvoiceDate'].max()
recency = max(recency)-recency
recency = recency.dt.days
recency = recency.rename("Recency")
```



As we can se, this recency **count the freshness of the customer activity**, in this case means time since last order.

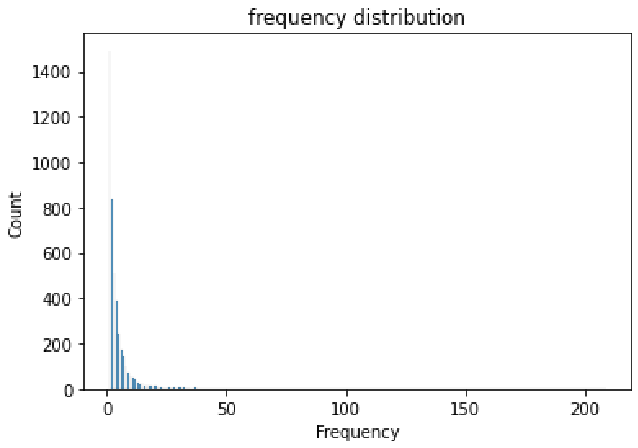
And we know, there is positive skewed distribution.

Frequency

```
freq = df.groupby(by=['CustomerID'])['InvoiceNo'].nunique()
freq = freq.rename("Frequency")

freq.head()
```

```
CustomerID
12346.0    1
12347.0    7
12348.0    4
12349.0    1
12350.0    1
Name: Frequency, dtype: int64
```



As we can se, this frequency **count the frequency of the customer transaction**, in this case means how many times customer repeat order.

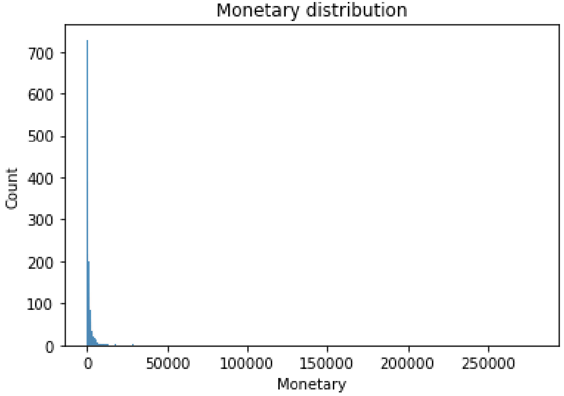
And we know, there is positive skewed distribution.

Monetary

```
monet = df.groupby(by=['CustomerID'])['TotalAmount'].sum()
monet = monet.rename("Monetary")

monet.head()
```

```
CustomerID
12346.0    77183.60
12347.0     4310.00
12348.0     1797.24
12349.0     1757.55
12350.0      334.40
Name: Monetary, dtype: float64
```



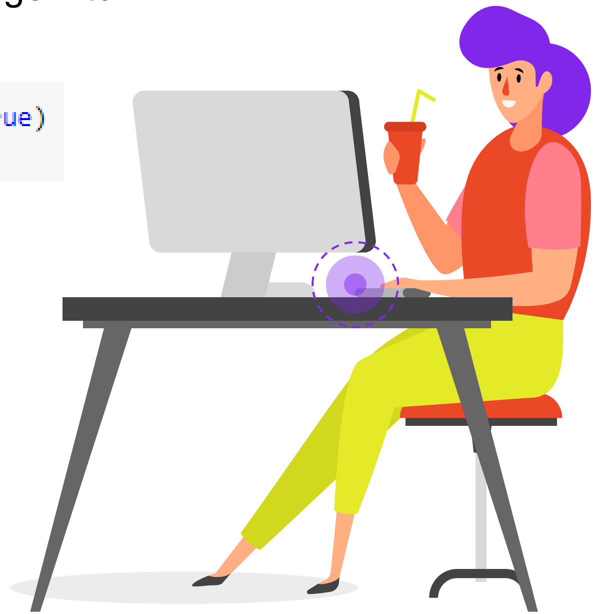
As we can se, this monetary shows **the intention of the customer's spend their of their purchasing power**, in this case means how much total customer spend order product.

And we know, there is positive skewed distribution.

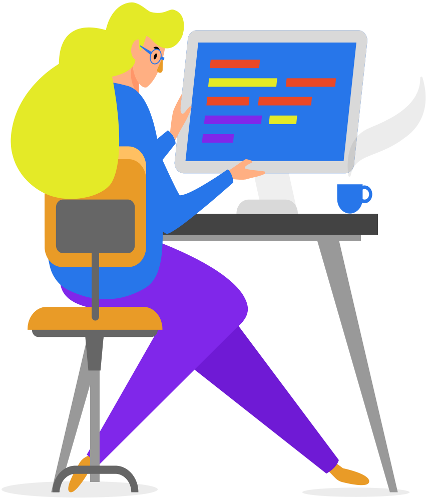
Categorical Data - Dummies

‘Country is UK’ using Dummies.
Because ‘Country is UK’ is still categorical data, then we should change into numerical data using Dummies.

```
df_new = pd.get_dummies(df_new, columns=['Country_isUK'], drop_first = True)
df_new = df_new.rename(columns = {'Country_isUK_UK' : "Country_isUK"})
```



Standart Scaler



Because the numeric columns have different scale, we need to scale it so all numerical features have different importance.

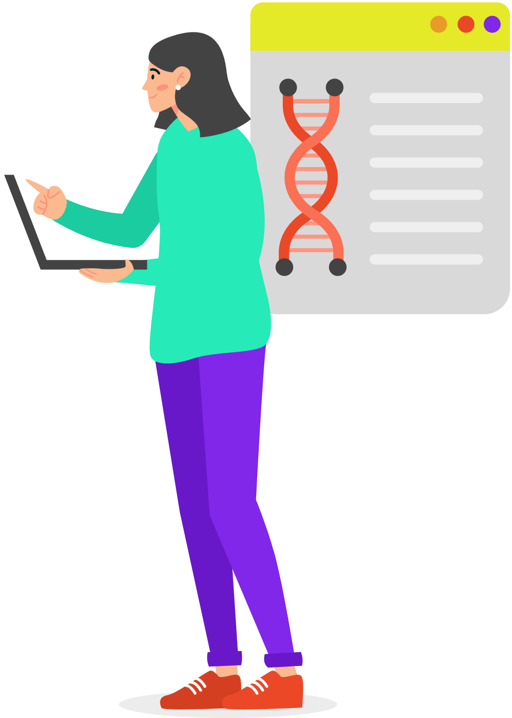
```
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_new)

df_scaled = pd.DataFrame(df_scaled, columns = df_new.columns, index = df_new.index)
```

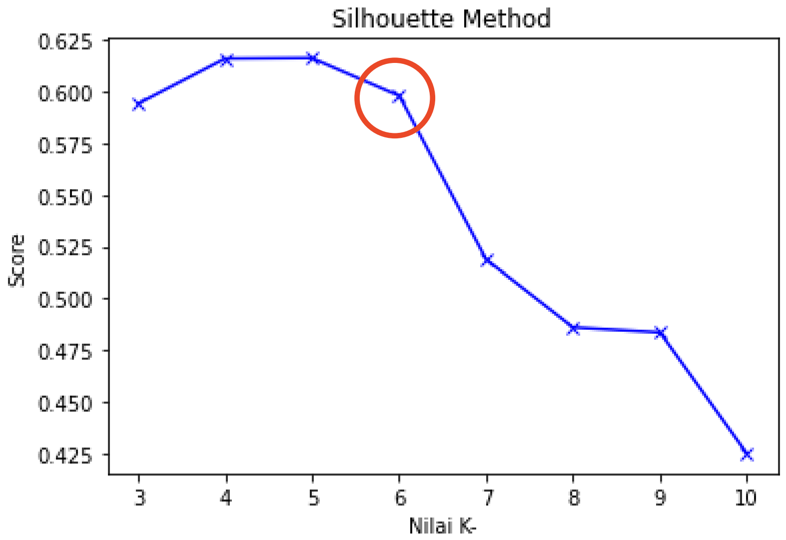
How The Data Looks Like before Cluster Analysis

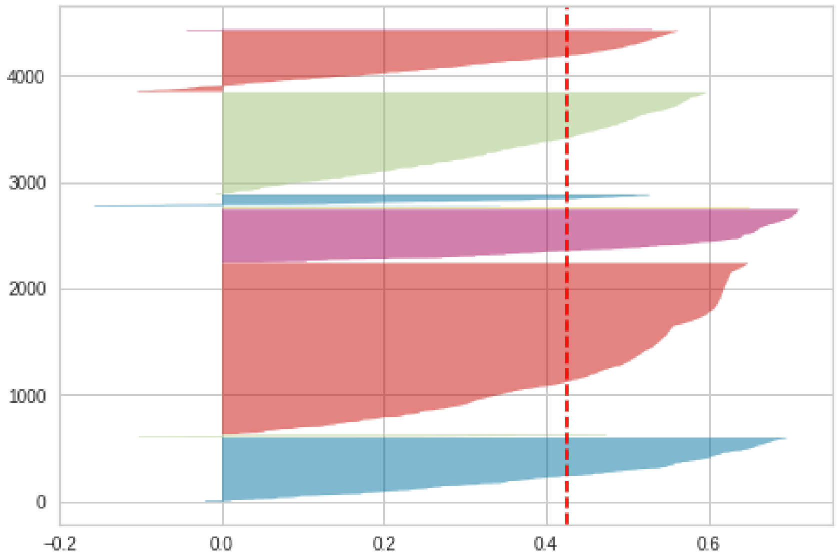
```
df_new.head()
```

	Recency	Frequency	Monetary	Country_isUK
CustomerID				
12346.0	325	1	77183.60	1
12347.0	1	7	4310.00	0
12348.0	74	4	1797.24	0
12349.0	18	1	1757.55	0
12350.0	309	1	334.40	0



Finding The ‘Right’ Number of Clusters





We can see, Based on Silhouette Analysis, Cluster number we can choose = **6**

Final Dataset before Customer Segmentation Analysis

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

Customer Segmentation

	CustomerID	Country_isUK	InvoiceDate	TotalAmount	Recency	Frequency	Monetary	cluster
0	17850.0	UK	2010-12-01 08:26:00	15.30	371	34	5391.21	3
1	17850.0	UK	2010-12-01 08:26:00	20.34	371	34	5391.21	3
2	17850.0	UK	2010-12-01 08:26:00	22.00	371	34	5391.21	3
3	17850.0	UK	2010-12-01 08:26:00	20.34	371	34	5391.21	3
4	17850.0	UK	2010-12-01 08:26:00	20.34	371	34	5391.21	3
5	17850.0	UK	2010-12-01 08:26:00	15.30	371	34	5391.21	3
6	17850.0	UK	2010-12-01 08:26:00	25.50	371	34	5391.21	3
7	17850.0	UK	2010-12-01 08:28:00	11.10	371	34	5391.21	3
8	17850.0	UK	2010-12-01 08:28:00	11.10	371	34	5391.21	3
9	17850.0	UK	2010-12-01 09:01:00	11.10	371	34	5391.21	3

Closing Thought

	cluster	0	1	2	3	4	5
Recency	mean	44.187713	2.000000	248.036862	14.662539	0.000000	21.333333
	std	36.455582	3.366502	65.848940	28.686906	0.000000	70.187843
	min	0.000000	0.000000	144.000000	0.000000	0.000000	0.000000
	q25	15.000000	0.000000	190.000000	2.000000	0.000000	1.000000
	median	34.000000	0.500000	242.000000	8.000000	0.000000	3.000000
	q75	66.000000	2.500000	300.000000	18.000000	0.000000	7.000000
	max	157.000000	7.000000	373.000000	371.000000	0.000000	325.000000
Frequency	mean	3.308532	45.250000	1.551040	16.931889	205.000000	54.142857
	std	2.309531	30.869348	1.072466	7.279364	5.656854	30.302287
	min	1.000000	2.000000	1.000000	3.000000	201.000000	1.000000
	q25	1.000000	35.000000	1.000000	12.000000	203.000000	31.000000
	median	3.000000	53.000000	1.000000	15.000000	205.000000	50.000000
	q75	5.000000	63.250000	2.000000	20.000000	207.000000	63.000000
	max	11.000000	73.000000	12.000000	55.000000	209.000000	124.000000
Monetary	mean	1211.113066	225721.652500	518.357534	7751.478669	88772.395000	58584.063810
	std	1306.689700	52818.123796	1495.826978	6646.067708	77856.225488	29255.539222
	min	6.200000	168472.500000	3.750000	1296.440000	33719.730000	11189.910000
	q25	372.910000	188031.217500	169.612500	3992.400000	61246.062500	37153.850000
	median	778.250000	227104.045000	309.670000	5591.420000	88772.395000	58510.480000
	q75	1601.882500	264794.480000	537.877500	8863.610000	116298.727500	66653.560000
	max	13219.740000	280206.020000	44534.300000	50491.810000	143825.060000	124914.530000

Recomendation

01 My recommendation for Rachel’s ecommerce is to **prioritize** this three customer segments, specially **Cluster 2, Cluster 4, and Cluster 5**

02 **Cluster 2** generate the highest purchasing power, with the average purchase power is in 225721.65

03 **Cluster 4** generate the most customer activity since their last order, with the average activity is in 15

04 **Cluster 5** generate the most customer transaction (repeat order), with the average activity is in 6

05 My recommendation for Rachel’s ecommerce is to **pay more attention to** this three customer segments, specially **Cluster 1, and Cluster 3.** Because customers in this cluster have the highest probability of not buying products at Rachel's ecommerce, by giving discount or free shipping

THANK YOU