**(Summary) Project 1: RFM Analysis with UCI online retail data**

20229007 Sungyoung Ryu (BTM539) AI for Business Management

**To refine the dataset before analysis:**

* At the first, the row data set from UCI has 525,461 instances from 2009 to 2010 and 541910 instances from 2010 to 2011, so used concat from pandas library to combine into one sheet as ‘btm539\_rfm\_analysis.csv’ (1,063,731)
* To exclude NaN or invalid data, get rid of #N/A (824,364 lefts) first, extract negative 'Q' (Return goods) and zero 'P' (Free gift maybe, but minus value of price can exist due to returned goods). After the assimilation, data has 805,549 instances.

**R, F, M Analysis:**

The purpose of 1st project is finding the correlation among the group all customers who bought some goods in 2009-2011 from UK online retailers. According to given guideline, I have to slice the data as R, F, M values from 3 variables:

* From this project to conduct RFM analysis with UCI online retail II data, Customer ID as key attributes. I set the recency(R) variable as date difference between InvoiceDate from next purchase to previous one. Also, frequency(F) variable is how many times did they purchase, and the last Monetary value(M) variable as products price by quality.
* These variables might be useful to how did consumers spent their money usually, frequently and how much did they spent. Also we can clustered 3 groups in each dimensions:
* Sorted as: ‘3’ means head group(most), ‘2’ means body group(middle), ‘1’ means tail group(least) – and can be made 27 combination (3 × 3 × 3) of customer groups.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1. Customer ID is key to identify the customer as unique value.  2. As a result, RFM\_Group attribute (111, 112, … 333) can be observed. Explain the characteristics, 111 is least contributed group and once they bought, they almost not to come back, otherwise, 333 is consist of high loyalty customers  3. In this way, now we are ready to analyze which groups of recent, frequent and heavy buyers contribute to the aggregated sales. | | | |
| Recency(r) | | R₁ | R₂ | R₃ |
| Frequency(f) | | F₁ | F₂ | F₃ |
| MonetaryValue(m) | | M₁ | M₂ | M₃ |

**Output:**

The density graph of ‘Recency’ displays two peaks, with the highest concentration observed around the very first point and another peak at 400. One online shopping enthusiast made 398 purchases, while most transactions occurred between 1 and 18 times. On ‘Monetary Value’ average, customers spent around 439, but one consumer placed an order worth £57,753.

텍스트이(가) 표시된 사진

자동 생성된 설명

**Limitation & lesson learnt:**

Even I got helps of many open source codes from GitHub, Reddit or Medium pages, I had struggled to how should I get what I want to adapt on this analysis.

Otherwise, tried to make a scatter plot for 27 customer groups, but outcome was little bit odd and every result isn’t going that I want to be. So in the end I used bar chart to explain the aggregated sales. Developer thinking is not implanted on my own neuron-synapse circuit yet.

Bad laptop & free version Colab limitation: I used the virtual environment for Google Colab, but sometimes lost the previously set environment. Due to architecture of Windows on ARM64x, no choice without using Colab. Because Windows on ARM(windows tablet such as MS Surface Go) doesn’t have any CPU, operating system relied on GPU like Qualcomm Snapdragon itself: When I did ML with large scale of dataset, sometimes it has limitation, stopped for a long often or error occurred in VSC or Python Idle. (One of my K-School colleagues told me it wouldn’t be matter because Mac used ARM64 architecture too.)

During winter break, I took a basic Python IT academy course for 2 weeks and I guess I spent all I learnt at there in this project. Also participated SAS bootcamp hosted by Prof. Kevin Kim(My advisor), SAS UNIX Whartons computing works better than personal computers, so if Colab do works like that, without hesitation to use.

Still 2nd, 3rd and final project left. Figure out the alternatives, probably Ubuntu implementation on WSL would solve the problem I hope… or I can try to use virtual computing access to desktop computer instead of my own Windows tablet.