Unsupervised Learning Competition

Mitchell Hughes - 19mh22
Bote Jiang - 13bj7
Henry Van Herk - 15hgvh

Python was the main language used for this competition. The code was written in a Jupyter notebook. The libraries used include: Pandas, Numpy, Sklearn, matplotlib, and the other basic system libraries that come with Pythonic installers. Python can be downloaded here https://www.python.org/downloads/ and Jupyter notebook can be downloaded here https://jupyter.org/install.

Introduction

With the proliferation of e-commerce, transaction data has become more transparent and accurate to the vendor. Information can include item information, customer information, quantity and price of purchase.

With this information, RMF (Recency, Frequency, and Monetary) analysis can be performed to assess a customer's value to the company. Recency tracks the most recent purchases for a customer, Frequency tracks how often a customer purchases, and Monetary tracks how much the customer spends.

This competition takes a dataset from an online retail company and performs clustering using RFM metrics. The customers will be segmented into different groups which will provide insight as to which customers are the most valuable.

Data Exploration

First we look at the number of unique values. This gives us insight as to which attributes or rows to remove:

We can see that each variable has duplicates (non-unique values). The obvious attributes include quantity and country. Stock code, description and customerID have similar number of unique values, therefore one might be redundant. Invoice number and date all have much more unique values but both have much less than the total number of entries. This tells us that they are not unique for each entry and can be potentially grouped.

```
[{'InvoiceNo': 25900,
  'StockCode': 4070,
  'Description': 4223,
  'Quantity': 722,
  'InvoiceDate': 23260,
  'CustomerID': 4372,
  'Country': 38}]
```

We can also look at the amount of missing data:

Since we are trying to segment customers into different groups. We can drop all of the rows with missing customer ID.
As a result, the number of missing description was also eliminated.

```
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID    135080
Country            0
```

We can try to determine the importance of the Description column. As seen below, description is more or less a duplicate of stock code.

```
InvoiceNo              object
StockCode              object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
```

There are more unique values in description because certain descriptions may slightly vary at different times. We can assume that each description corresponds with a unique stock code, therefore we can drop it.

We can also analyze the negative quantity values.

These can be useful monetary analysis however should be removed for frequency and recency. There are records that are not purchases, some are for negative quantities, indicating they are returns/canceled orders. These records should not be included in the analysis, but we have to make a decision as to what to do with the orders that are being cancelled. In some cases there is only one pair of StockCode, CustomerID (eg one purchase which was later returned), which would suggest that the return and original purchase should both be removed. In other cases, a customer may have purchased the item multiple times and/or in a higher quantity that is being returned (eg. purchases 6 units, returns 2 or purchases 1 unit 6 times, then returns 2). In this case, subtracting the monetary value of the return from the customer's total expenditure is necessary. So we will get the net purchase value per customer, remove the transactions with negative quantities, then construct the frequency and recency measures from the original purchases.

```
online.Quantity.min()
```

```
-80995
```

```
online[online['Quantity'] < 0 ].head(n=5)
```

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom
236	C536391	21984	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom

Data Engineering

Need to make statistics for each customer ID:

- Recency (date since last transaction)
- Frequency (how often purchases are made)
- Monetary (value of spending per customer)

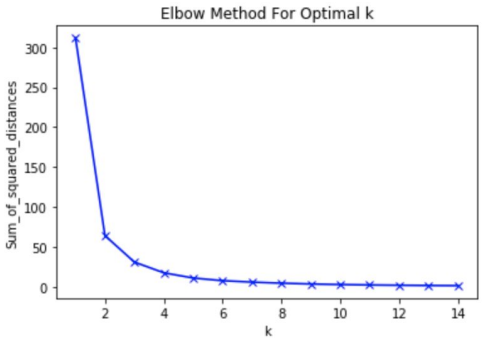
We do monetary analysis first as we need to keep the negative quantity. The monetary value per customer is calculated by: Quantity x Amount and summed for all transactions. The returns and cancellations should cancel out with the positive values. For Recency, we find the difference between the last date as presented in the competition description and last transaction per customer. For Frequency we sum all of the invoice counts for each customer. Recency and Frequency are calculated on a copy of the dataframe with the negative quantities removed. We finally left join all three to form a table with CustomerID, Recency, Frequency, and MonetaryValue.

We then rank each of the R, F and M attributes to a scale of 5 using 5 equal quantiles. Each individual score is then summed to create a combined score. The combined score is then discretized into “best”, “medium”, “worst”.

	CustomerID	Recency	Frequency	MonetaryValue	R_rank	F_rank	M_rank	Combined_rank	Combined_rank_Labeled
0	12346.0	0.871314	0.000000	0.004152	1	1	1	3	Worst
1	12347.0	0.005362	0.023069	0.019509	5	5	5	15	Best
2	12348.0	0.201072	0.003824	0.010556	2	3	4	9	Medium
3	12349.0	0.048257	0.009177	0.010414	4	4	4	12	Best
4	12350.0	0.831099	0.002039	0.005344	1	2	2	5	Worst

Clustering

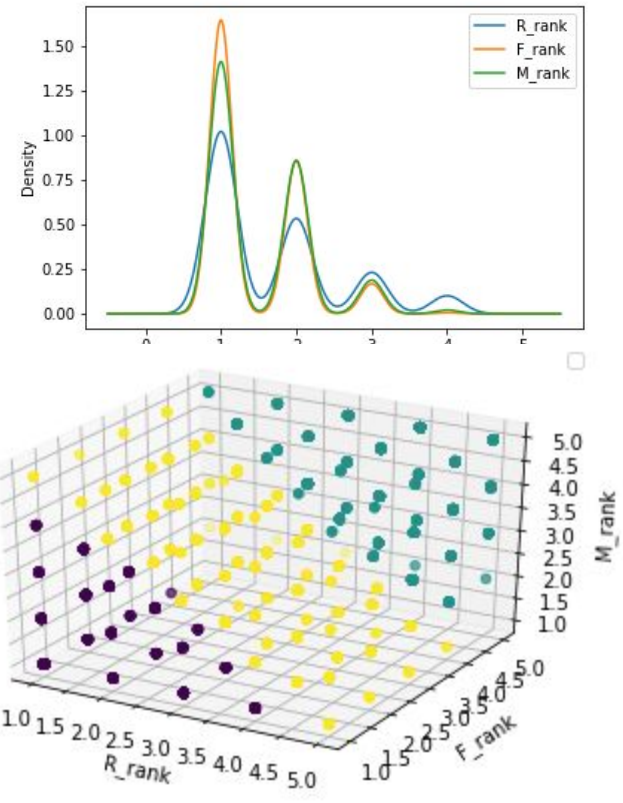
We first do normalization. Normalization makes training less sensitive to the scale of features, so we can better solve for coefficients. We then determine the number of clusters needed using the Elbow method. The idea is to choose a small number of clusters that also has low sum-squared-error. We do this by finding the “elbow” of the plot. In this case k should equal to 3.

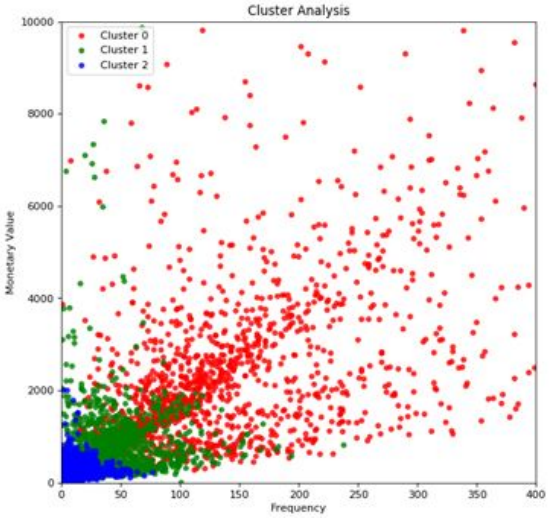


We finally do k-means clustering on ['Recency','Frequency','MonetaryValue'] with k = 3 and a max iteration of 1000. We also set a constant starting seed to obtain the same clustering naming every time the code is executed.

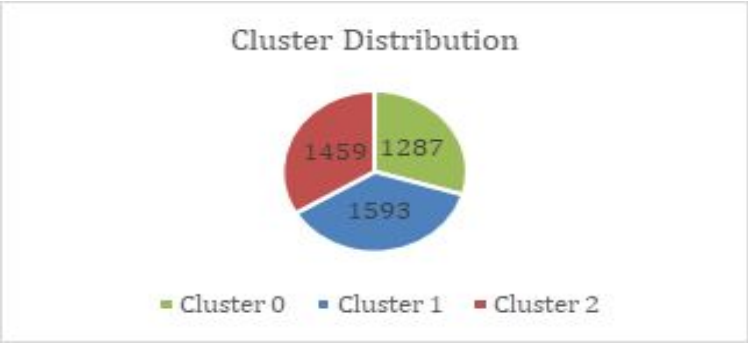
Analysis

Looking at the makeup of each identified cluster, it appears that there are clear groupings of low, medium and high value customers.
The first cluster is mainly composed of records with scores of 1 and 2, and importantly few scores of 4 and no scores of 5. This cluster will be referred to as the low value group. 54% of the lowest ranked recency records appear in this cluster, 61% and 57% of the lowest ranked Frequency and Monetary ranks are also in this cluster. Similarly, the highvalue group contains all of the records which score 5 in more than one category. The clustering separates the customer’s into well defined clusters, with no ambiguous points.





When we visualize all three clusters' monetary value against their frequency in a scatter plot, we can identify a linear correlation between the two attributes. In what seems to be most cases, increased monetary value is related to increased frequency. In terms of the cluster separations, cluster 0 (red), is the least intra-related of the three with the highest ranging monetary values and frequencies. Cluster 1 (green), is more intra-related than cluster 0 (red) with a lower range of monetary values and frequencies. Cluster 2 (blue), is the most intra-related of the three clusters and has the smallest range of monetary values and frequencies. In terms of inter-relationships between the clusters when comparing their monetary values and frequency, cluster 1 (green) and cluster 2 (blue) are moderately inter-related, a sizeable portion of cluster 1 (green) covers almost the entire scope of cluster 2 (blue). Additionally, there is a low level of inter-relationship between cluster 0 (red) and cluster 1 (green), and cluster 0 (red) and cluster 2 (blue), although cluster 0 (red) covers a vast majority of the scope of clusters 1 (green) and 2 (blue), since it's monetary value and frequency vary so much in comparison, we can not say they are strongly interrelated.



The distribution of the customers within each cluster is almost even. Cluster 0 (green) has 1287 of 4339 total customers, or 29.66% of total customers, cluster 1 (blue) has 1593 total customers, or 36.71% of total customers, and cluster 2 (red) has 1459 customers, or 33.63% of total customers.

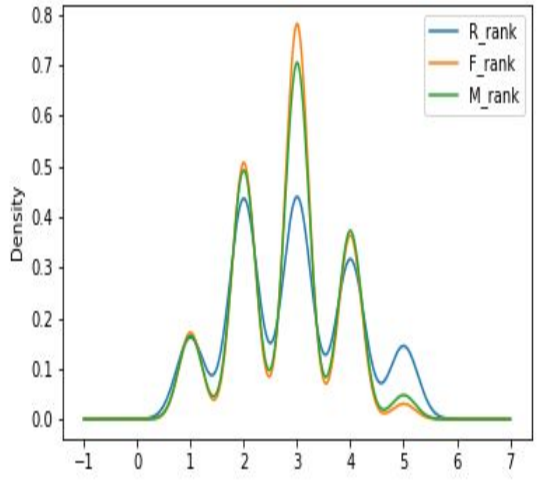
The distinct feature of cluster 0 is that it is possible that it represents old customers with no recent purchases (low recency, high frequency and medium monetary), with a low median recency, a high median frequency, and a low monetary value.

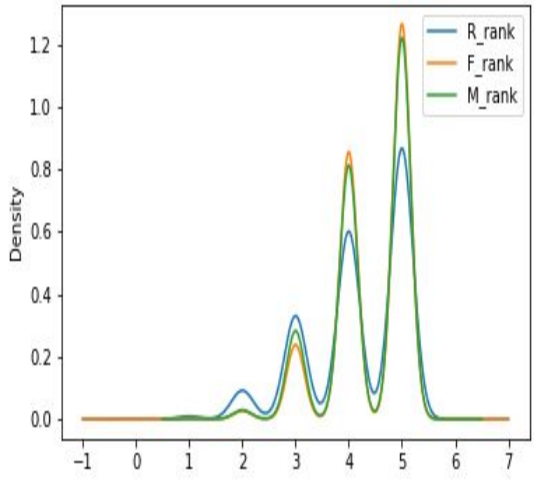
The distinct feature of cluster 1 is that it is a contender for the loyal (most frequent) group, with the highest median frequency of 127.0. The distinct feature of cluster 2 is that it represents the potential highly profitable customers (recent and medium monetary), it has a medium recency, and the second largest median monetary value.

When comparing all clusters to try and find the highest and lowest revenue groups, a few things standout. If we rank based on monetary sum, cluster 0 makes up 3.9% of revenue, cluster 2 makes up 13.4% of total revenue and cluster 1 makes up 82.6% of total revenue. When we account for how frequency, cluster makes up 4.6% of all purchases, cluster 2 makes up 15.7% of all purchases and cluster 1 makes up 79.6% of all purchases. Although this makes it seem as though cluster 1 is the highest spending customer group by far, cluster 2 makes an average of 17.80$ per transaction, cluster 0 makes an average of 17.83$ a transaction, and cluster 1 makes an average of 21.67$ per transaction. So, despite making up 82.6% of total revenue and 79.6% of all purchases, the high value cluster is not that much greater than it initially seemed. Overall, cluster 0 is the lowest revenue group, and cluster 1 is the highest.

Appendix

Density for Medium Value cluster and High value cluster





Summary statistics by cluster

Cluster 0	Recency	Frequency	Monetary
Sum	241743	18329	3.27×10^4
Min	13	1	-1165.3
Median	191.0	12.0	215.08
Max	372	77	2002.4

Cluster 1	Recency	Frequency	Monetary
Sum	37616	316964	6.87×10^6
Min	-1	1	249.72
Median	15.0	127.0	2089.85

Max	371	7847	279489.02

Cluster 2	Recency	Frequency	Monetary
Sum	115670	62631	1.115×10^6
Min	-1	1	0.2664
Median	57.0	36.0	600.39
Max	372	238	21535.9