CISC451 Competition 3 Report

Group Member:
Steven Wen 20144322 Feiting Yang 20143750 Yifan Zhu 20146990

Package used:

Pandas: We use it to do fundamental data analysis. To use pandas, we just need to simply install using: pip install pandas, and then import pandas in python code.

Numpy: We use it to do mathematical calculations. To use it, we should use the 'pip install numpy' instruction and then import it into python code.

Sklearn: We use sklearn's different package for data processing and modeling. This package can be installed by 'pip install -U scikit-learn'.

Math: We used a math package to do math calculations like power. This can be implemented by 'import math'.

Mpl_toolkits.mplot3d: We use this package to graph 3d graphs. To import just 'import mpl_toolkits.mplot3d'.

KMeans: We use k means as our model. To import k-means we just "from sklearn.cluster import KMeans".

AgglomerativeClustering: Another clustering model we used. To complement we used 'from sklearn.cluster import AgglomerativeClustering'

Data Preprocessing:

Missing Data:

We find that there are two columns with missing data one is description and the other one is customer ID. For the description there are 1454 missing values, the percentage of the missing value is 0.26%. For customerID there are 135080 missing values which are 24.9% of the data. Considering both missing data are not a huge percentage, we decided to drop it. We also dropped the data that is "C" in the Invoice No, since it indicates a cancellation.

Group Data:

We grouped the data by the customer's ID since there are multiple orders from the same buyer, and calculates when it is the last time they purchased, frequency of the purchase and how much it purchased.

CustomerID	recency	frequency	monetary
12346.0	325	1	77183.60
12347.0	2	7	4310.00
12348.0	75	4	1797.24
12349.0	18	1	1757.55
12350.0	310	1	334.40
...	...	...	...
18280.0	277	1	180.60
18281.0	180	1	80.82
18282.0	7	2	178.05
18283.0	3	16	2094.88
18287.0	42	3	1837.28

Figure 1: Group data

Outlier:

We used the 'quantile' function in pandas to drop the lower and upper 25% as the outliers. Increased the BetaCV from 0.18 to 0.41.

Data Normalization:

To normalize the data we first used cube root to deal with the skewness which is a fairly strong transformation with a substantial effect on distribution shape. Then we applied standard deviation divided by the difference between data and the mean.

Model Implementation:

K-mean:

As this is a clustering problem, the first model that came to mind was k-means since it is used to find groups which have not been explicitly labeled in the data. We use k-means to group 4 clustering by calling the sklearn package. And set this as our baseline model. Later we find out that 5 clustering is better suitable for dividing differences in customers.We turned the model by changing the tol - Relative tolerance with regards to Frobenius norm. Then we evaluate this by using BetaCV which is the ratio between the average of intra-cluster distance to the average of inter-cluster distance. And we got 0.4125 as the result.

AgglomerativeClustering:

By trying to improve the k-means algorithm we find the agglomerative clustering which is an unsupervised machine learning technique that divides the population into several clusters such that data points in the same cluster are more similar and data points in different clusters are dissimilar. And to call in we simply used the sklearn package. We also performed the BetaCV to evaluate the cluster and got 0.4688 as the result which is worse than K-means.

Result:

Clusters:

By using mpl_toolkits.mplot3d we have output the following group which shows the 5 clusters.
The cluster 0 has a distribution of 22.4198%, cluster 1 has 18.5664%, cluster 2 has 24.6370%, cluster 3 has 18.4047%, and cluster 4 has 19.9407%. Calculated using customers per cluster divided by the total number of customers. And we can see the clusters from the following figure. **Yellow** as cluster 0, **orange** as cluster 1, **red** as cluster 2, **purple** as cluster 3 and **blue** as cluster 4.

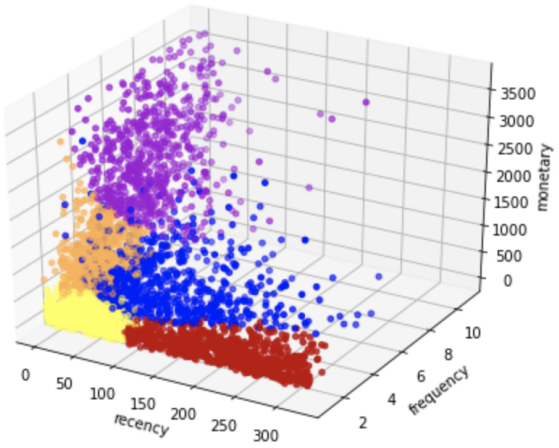

Figure 2: clusters

Statistics of each cluster

Min:

cluster	recency	frequency	monetary
0	1	1	0.00
1	0	1	201.12
2	101	1	3.75
3	0	2	694.40
4	30	1	70.02

Max:

cluster	recency	frequency	monetary
0	99	4	1013.01
1	60	7	3192.54
2	326	3	1063.00
3	264	11	3692.28
4	319	7	3528.34

Median:

cluster	recency	frequency	monetary
0	38.0	1.0	277.425
1	18.0	3.0	853.720
2	233.0	1.0	237.610
3	24.0	6.0	2124.990
4	106.0	2.0	801.500

Distinct features:

least profitable customer group: Red cluster 2. Haven't purchased recently, not a frequent buyer and monetary is not high.
most profitable customer group: Purple cluster 3. Having purchased recently, high frequency buyers and monetary is high.
The loyal (most frequent) group: Blue cluster 4 frequency is high and recency is also wide.
Old customers with no recent purchases (low recency, high frequency and medium monetary): Red cluster 2.
Potential highly profitable customers (recent and medium monetary): Orange cluster 1.