zhaochengyu netid = zyr531

1.Bussiness Target.

My target is to maximize the insurance profit of specific services furnished by specific Medicare providers by finding the most profit potential Medicare service in Illinois.

2.Intuitive Approach

There are 26 attributes of this dataset and 6 of them are numeric. I would like to add an attribute called Total charge paid which equals the product of Average Submitted Charge Amount and Number of Services. If the Total charge paid of a specific kind services is high, that is to say there is a high profit potential because people are willing to pay more to the insurance company for that specific kind of services to minimize their financial loss. And another attribute I want to mention here is Number of Medicare Beneficiaries. If the value of this attribute is high, that means there is a high risk that my company would pay for this kind of services.

Then the target is clear. The clustering algorithm should help me to find the most profit potential service that my company should focus on while at the meantime the risk of that services should be as low as possible. In other words, the higher the amount of Total charge paid and the lower the value of Number of Medicare Beneficiaries, the kind of services is the better.

3.Data Preprocessing

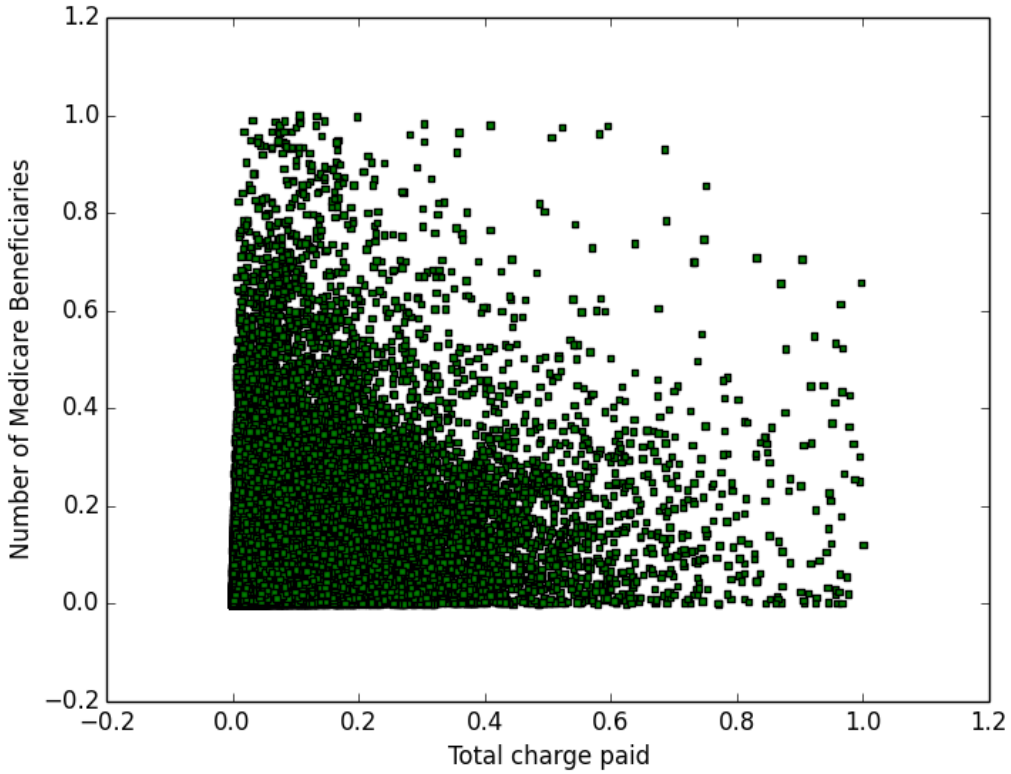
Among the whole dataset I choose those rows whose attribute ‘State Code of the Provider’ is ‘IL’. After dropping NAs, pairwise correlation is done in order to eliminate unnecessary data exploration. The results can be viewed below.(PS. NS = Number of Services, NMB = Number of Medicare Beneficiaries, NDMB = Number of Distinct Medicare Beneficiary/Per Day Services, AMAA = Average Medicare Allowed Amount, ASCA = Average Submitted Charge Amount, AMPA = Average Medicare Payment Amount, AMSA = Average Medicare Standardized Amount.)

	NS	NMB	NDMB	AMAA	ASCA	AMPA	AMSA
NS	1	0.43	0.52	-0.01	-0.01	-0.01	-0.01
NMB		1	0.86	-0.09	-0.10	-0.09	-0.09
NDMB			1	-0.09	-0.10	-0.09	-0.09
AMAA				1	0.75	-1.00	-1.00
ASCA					1	0.75	0.75
AMPA						1	0.99
AMSA							1

4.Methodolgy and Result

Outlier deletion, data normalization and data standardization have been done to make the data become cleaner, clearer and easier to visualize.

Figure below shows the 2 dimensional scatter plot of Total charge paid vs Number of Medicare Beneficiaries.



The dataset has 382620 rows. It takes a very long time to do hierarchical clustering. Because of the large dataset, it is difficult to identify the correct number of clusters by the dendogram. And therefore I chose k-means clustering method and choose the number of cluster by trial and error.

Since the dataset it really large , I have to take random sample in order to calculate silhouette score and draw silhouette figure effectively.

Figure below shows the results from n_clusters = 2 to n_ clusters = 5. The number of sample dataset is 10% of the whole dataset.

Figure 1 n_clusters = 2

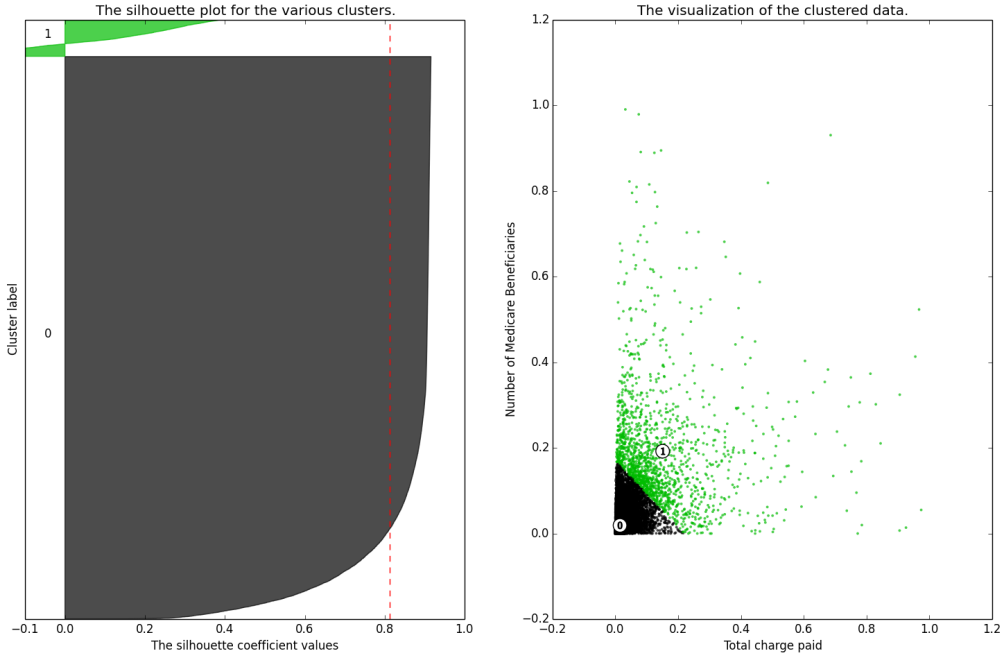


Figure 2 n_clusters = 3

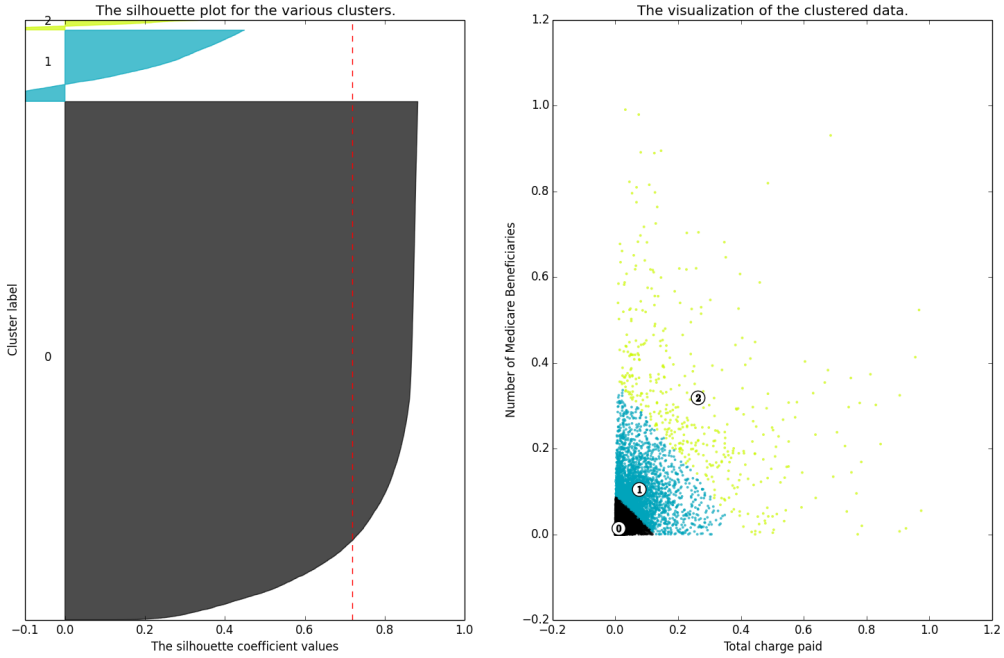


Figure 3 n_clusters = 4

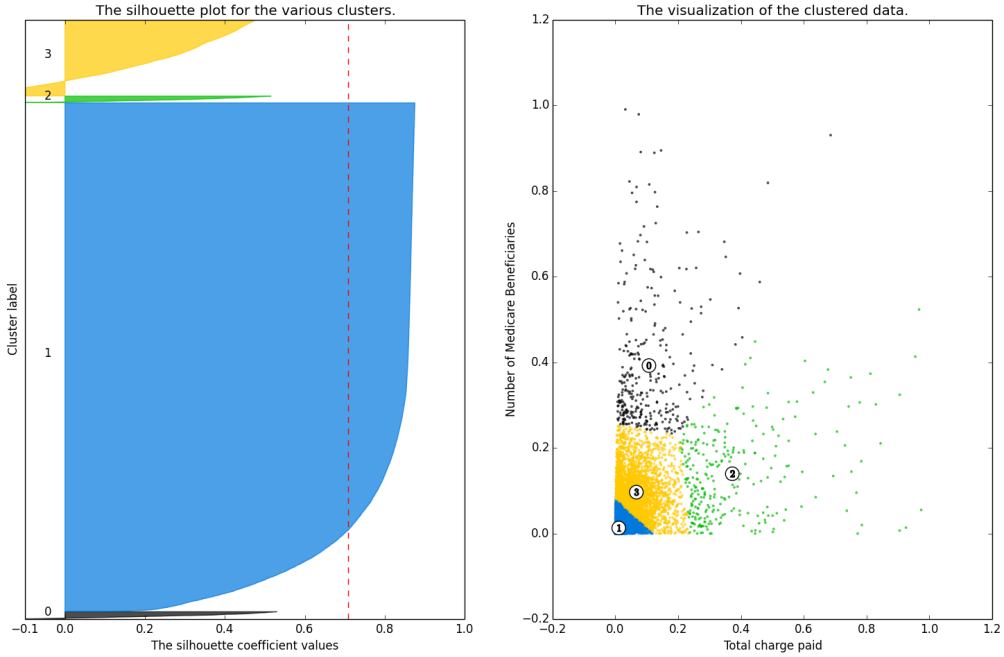
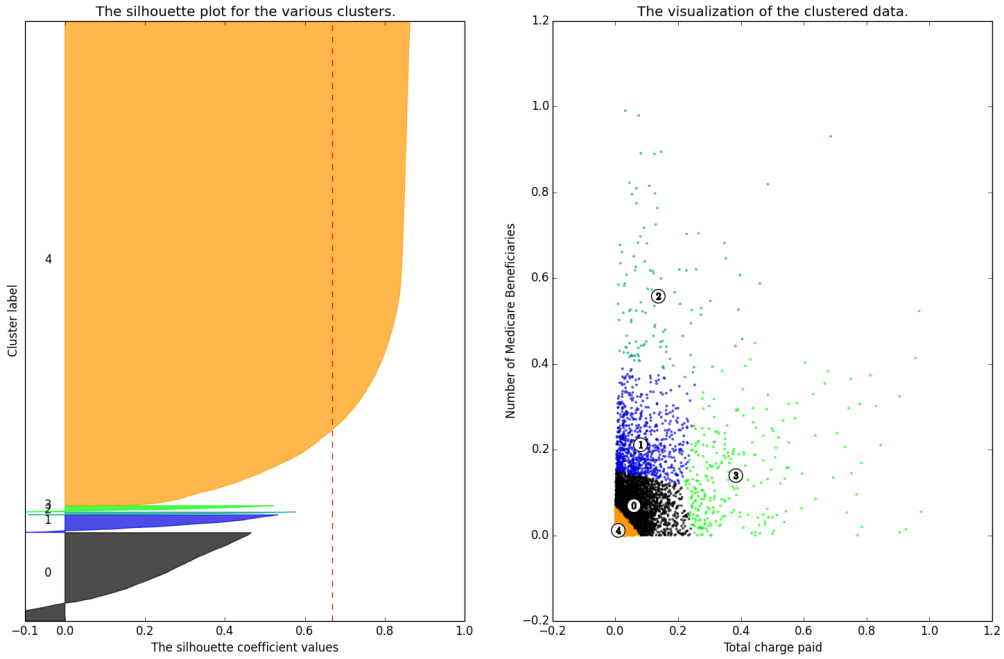


Figure 4 clusters = 5



Number of clusters	Silhouette score
2	0.8123
3	0.7190
4	0.7077
5	0.6681

5.Conclusion

Based on the silhouette score and business logic. I think the best clustering strategy is to get 4 clusters. According to the scatter plot *Figure 3* ,cluster 1 and cluster 3 are mediocre services whose profit potential and risk are normal. Services in cluster 0 are non-profit and high risky which are not worthy spending too much time. Services in cluster 2 are our target services which are highly profitable with relatively low risk. We should spend more time on those targets and find a good pricing strategy to maximize our profit.

The most serious problem is the large dataset that could not be fully used to calculate silhouette score, which means there might cause some accuracy loss of centroids' locations. Based on the result from k-means. It is not hard to find numbers of points in each cluster is heavily unbalanced, which means hierarchy clustering might be a better choice. However, when it comes to the relatively big dataset, the time complexity would be a disaster.

All source code are written in python and would be uploaded through github.