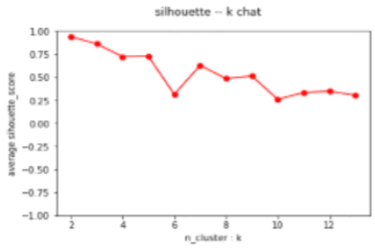
Question2 报告

一、运行结果

下图是程序在 jupyter 上的运行结果

```
In [5]: if __name__ == '__main__':
            data, data_array, vipno_num, vipno_len = pre_data()
            question_a(data_array, vipno_num)
            #可知当k=2时，silhouette的值最高
            question_b(data, data_array, vipno_num, vipno_len)
```

```
Initial k: 8
For n_clusters =  2 The average silhouette_score is : 0.940586344296
For n_clusters =  3 The average silhouette_score is : 0.86216444571
For n_clusters =  4 The average silhouette_score is : 0.723579173898
For n_clusters =  5 The average silhouette_score is : 0.726903182347
For n_clusters =  6 The average silhouette_score is : 0.314762191847
For n_clusters =  7 The average silhouette_score is : 0.628329266266
For n_clusters =  8 The average silhouette_score is : 0.485552562437
For n_clusters =  9 The average silhouette_score is : 0.513205360844
For n_clusters =  10 The average silhouette_score is : 0.259986673116
For n_clusters =  11 The average silhouette_score is : 0.334443201272
For n_clusters =  12 The average silhouette_score is : 0.348167787607
For n_clusters =  13 The average silhouette_score is : 0.30429823005
```




```
for n_cluster =  2 , hash_size =  2 , k =  1 , vipno_pos =  139 , knn =  [1593140598586]
output:  1593140598586
Same cluster

for n_cluster =  2 , hash_size =  2 , k =  2 , vipno_pos =  139 , knn =  [1596140797485, 1590151544861]
output:  1596140797485
Same cluster
output:  1590151544861
Same cluster

for n_cluster =  2 , hash_size =  2 , k =  3 , vipno_pos =  139 , knn =  [1593140598586, 1591020667889, 1590142175272]
output:  1593140598586
Same cluster
output:  1591020667889
Same cluster
output:  1590142175272
Same cluster

for n_cluster =  2 , hash_size =  2 , k =  4 , vipno_pos =  139 , knn =  [1596140797485, 1591020667889, 1595132332932, 1590142192491]
output:  1596140797485
Same cluster
output:  1591020667889
Same cluster
output:  1595132332932
Same cluster
output:  1590142192491
Same cluster

for n_cluster =  2 , hash_size =  2 , k =  5 , vipno_pos =  139 , knn =  [1596140797485, 1593140967467, 1590151470542, 1598140121611, 159014
2516563]
output:  1596140797485
Same cluster
output:  1593140967467
Same cluster
output:  1590151470542
Same cluster
output:  1598140121611
Same cluster
output:  1590142516563
Same cluster

for n_cluster =  2 , hash_size =  14 , k =  1 : no knn output
for n_cluster =  2 , hash_size =  14 , k =  2 : no knn output
for n_cluster =  2 , hash_size =  14 , k =  3 : no knn output
for n_cluster =  2 , hash_size =  14 , k =  4 : no knn output
for n_cluster =  2 , hash_size =  14 , k =  5 : no knn output
for n_cluster =  2 , hash_size =  29 , k =  1 : no knn output
for n_cluster =  2 , hash_size =  29 , k =  2 : no knn output
for n_cluster =  2 , hash_size =  29 , k =  3 : no knn output
for n_cluster =  2 , hash_size =  29 , k =  4 : no knn output
for n_cluster =  2 , hash_size =  29 , k =  5 : no knn output
for n_cluster =  2 , hash_size =  59 , k =  1 : no knn output
for n_cluster =  2 , hash_size =  59 , k =  2 : no knn output
for n_cluster =  2 , hash_size =  59 , k =  3 : no knn output
for n_cluster =  2 , hash_size =  59 , k =  4 : no knn output
for n_cluster =  2 , hash_size =  59 , k =  5 : no knn output
for n_cluster =  2 , hash_size =  89 , k =  1 : no knn output
for n_cluster =  2 , hash_size =  89 , k =  2 : no knn output
for n_cluster =  2 , hash_size =  89 , k =  3 : no knn output
for n_cluster =  2 , hash_size =  89 , k =  4 : no knn output
for n_cluster =  2 , hash_size =  89 , k =  5 : no knn output
for n_cluster =  2 , hash_size =  149 , k =  1 : no knn output
for n_cluster =  2 , hash_size =  149 , k =  2 : no knn output
for n_cluster =  2 , hash_size =  149 , k =  3 : no knn output
for n_cluster =  2 , hash_size =  149 , k =  4 : no knn output
for n_cluster =  2 , hash_size =  149 , k =  5 : no knn output
```

二、讨论分析

K-Means 聚类算法的主要步骤是：

1. 随机在取 K 个中心点。
2. 然后对图中的所有点求到这 K 个中心点的距离，假如点 Pi 离中心点 Si 最近，那么 Pi 属于 Si 点群。
3. 更新中心点，我们要移动中心点到属于他的“点群”的中心。
4. 重复第 2）和第 3）步，直到，中心点没有移动

第二题要求通过求聚类的 silhouette 系数来判断聚类的优劣，并找出聚类效果最好时中心点 k 的值。silhouette 系数范围是(-1,1)，系数越大，说明聚类的效果越好。通过绘制 silhouette-k 曲线图，初始 k = 8，不妨假设 k∈（2， 14）。如上图所示，当 k = 2 时，silhouette 系数取到最大值。可知，当簇的个数为 2 个时，聚类的效果最好，这和第一题得到的结果一样，即：采用的数据具有两个主要特征。

当 k = 2 时，将第一题得到的 knn 结果一一比对，观察其结果和输入是否是落到了 K-Means 算法所分出的同一个簇里面。如上图所示，取随机的 vipno 作为输入，从结果可以看到几乎所有的输入和输出都在同一个簇中。

K-Means 算法本身具有一些缺陷，其 K 值必须是事先给定的，很多时候，我们都不知道这个值应该取多少才合适，可能需要浪费很多时间才能找到合适的结果。在和第一题的结果进行比对之后发现，K-Means 所得到的结果和局部敏感哈希在很大程度上吻合，所以不妨用局部哈希得到的结果来对 K 值进行一个预测，这样能避免中心点难以确定的缺陷。

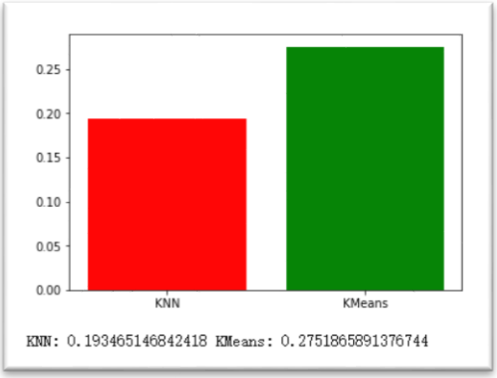
为了进一步确认，我将第二问所生成的聚类标签打印出来如下：

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0]
```

可以看到虽然分了两类，但几乎所有的都被分到了一个簇中，但是此时 silhouette 系数很高，不妨做出假设，数据源大部分数据都是相似的，只有特别少部分数据具有不同的特征，而这些小部分的数据，与其将它们分为一个簇，不如将其视为噪点。

三、性能

下图是所用时间的柱状图 KNN 的时间是指每次查询的时间，KMeans 是指聚类过程所用时间：



KNN: 0.193465146842418 KMeans: 0.2751865891376744