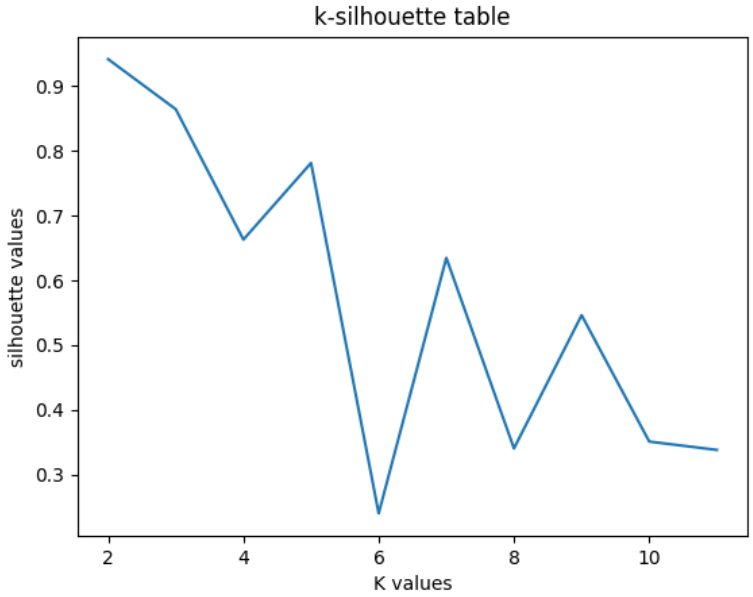
Q2 报告

1. 代码运行截图

a) Silhouette 系数值-k 值的函数图



b) 验证 Q1 中 lsh 的 knn 查询结果是否与本程序输入 vipno 所在同一个簇：

```
D:\Anaconda3\envs\ml3.5\python.exe C:/Users/Stephen/Desktop/数据挖掘/hw1/hw1/q2/b/verify.py
输入的vipno是1591015088262
其桶中的vipno有：
1591015088262
1595151630507
2900002512281
1591016151613
1590142192491

lsh的knn分类后的某一个桶中vipno所对应的矩阵索引值：
84
98
239
43
203

标签为84的分类为：1
标签为98的分类为：1
标签为239的分类为：1
标签为43的分类为：1
标签为203的分类为：1

Process finished with exit code 0
```

2.讨论分析部分

2.1 图表说明

该图表横坐标为 K-Means 算法中 K 的值。在 scikit-learn 中，Kmeans()的函数有 n_clusters 系数（即 k 的值，意味着被分成聚类的个数）。根据调整 k 的值（2 到√n/2），可以得到不同的聚类结果。将聚类结果利用 silhouette 系数进行评价可得到 average score，即聚类算法效果的好坏。效果越好系数越高。

由图可见，当 k 为 2 的时候，聚类效果最好。

2.2 相关说明

2.2.1 K-Means 算法

k-means 算法将一组样本 X 划分为 K 个不相交的聚类 C，每个聚类 C 由聚类中样本的均值描述。算法步骤是：

1. 从 N 个数据中任意选出 k 个数作为种子，作为 k 个簇的代表点
2. 剩下的点依此算出与这 k 个簇的中心的距离，并选取距离最短的那个簇作为这个点所属的簇
3. 重新计算每个簇的中心距离，重复上述步骤

缺点：此算法不适合凸型数据的 cluster 分类，同时不能处理噪音值。

2.2.2 聚类评估算法

轮廓系数（Silhouette Coefficient），是聚类效果好坏的一种评价方式。它结合了内聚度和分离度两种因素，可以用来在相同原始数据的基础上用来评价不同算法、或者算法不同运行方式对聚类结果所产生的影响。评价的具体方法为：

1，计算样本 i 到同簇其他样本的平均距离 a(i)。a(i)越小，说明样本 i 越应该被聚类到该簇。将 a(i)称为样本 i 的簇内不相似度。

2，计算样本 i 到其他某簇 C(j) 的所有样本的平均距离 b(i,j)，称为样本 i 与簇 C(j) 的不相似度。定义为样本 i 的簇间不相似度：b(i) =min{b(i, 1), b(i, 2), ..., b(i, k)}

3，根据样本 i 的簇内不相似度 a(i)和簇间不相似度 b(i)，定义样本 i 的轮廓系数为：

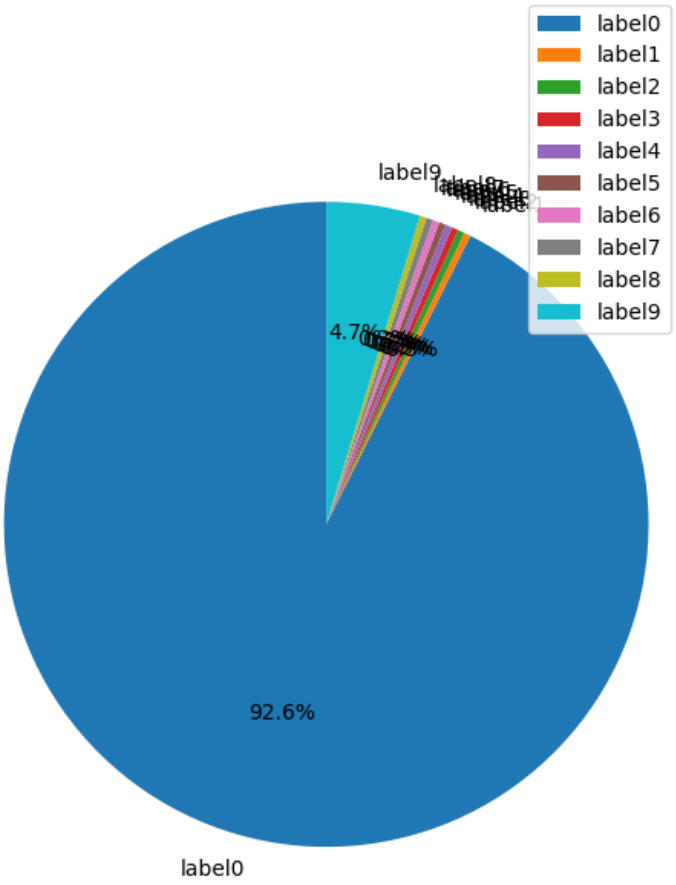
$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

4. si 接近 1，则说明样本 i 聚类合理；si 接近-1，则说明样本 i 更应该分类到另外的簇；若 si 近似为 0，则说明样本 i 在两个簇的边界上。

2.2.3 效果评估

首先看折线图。折线图表现出来的特征是：大体趋势下降，震荡比较明显。由图可得出：数据的聚类特征并不是特别明显；若数据可聚类的特征较为明显（可被分成 N 类（N>2））的话，则会在 K 取 N 的时候取到最大值；而且因为聚类特征明显，并不会出现较大的 k-silhouette 图表的震荡。

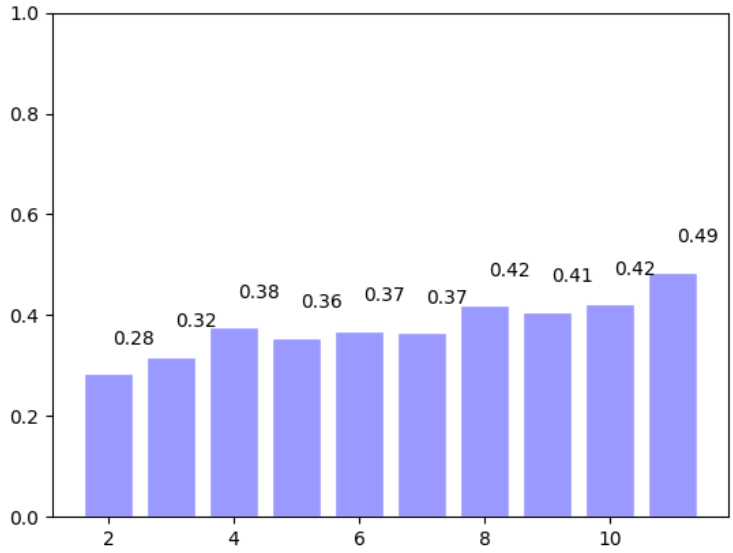
取 silhouette 系数最大时的 k 值进行测试（k = 2），得出分类结果为：297 个元素属于一个簇，只有一个元素属于另一个簇；而 k 取 10 的时候：



绝大部分数据被分到了一起这并不是好的聚类结果。因此可判断簇的个数>2 的时候可聚类性不强。

然后看 b 小问的输出结果。经过 lshash 选出 vipno 为 1591015088262 所在桶的所有 vipno，然后将其在矩阵中的索引值得到，分别查询在 K-Means 算法下得出的 label 值。经过多次测试，发现 label 值均一样，说明 lshash 进行的相似化归类结果符合 K-Means 聚类结果。结合上述折线图得到的信息，可判断绝大部分的数据被分到了一起，而从桶中随机取到的 vipno 大概率是这些数据中的一个，被分到和它一个桶中的数据也就大概率在 K-Means 中被分到了一起。

3. 性能图表



此图表为 k-time 柱状图。由图表可以看出，随着 k 值的不断增长，其运算时间也在逐渐增长，但涨幅并不是很大