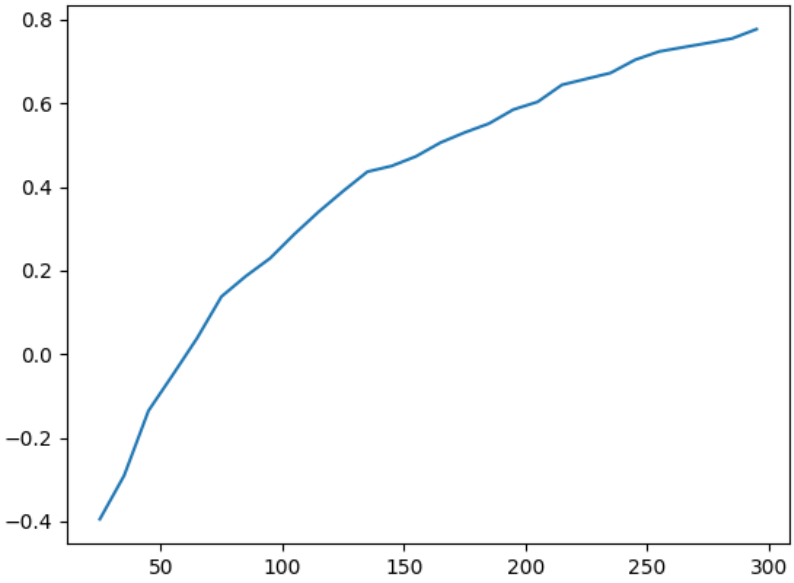
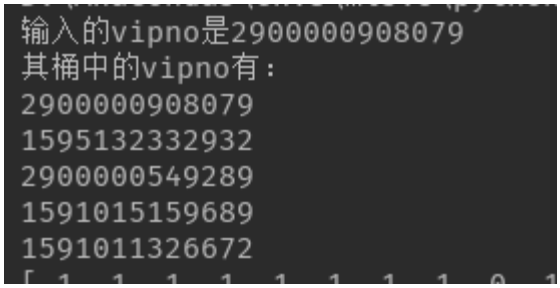
Q3 报告

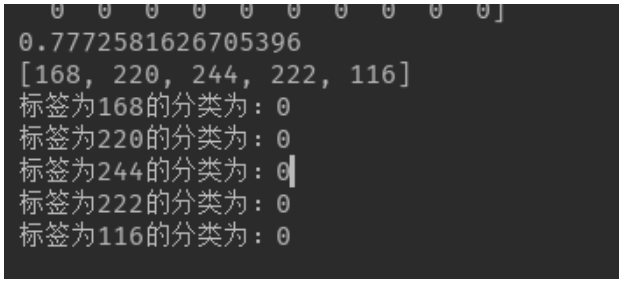
1. 代码运行截图

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：





2.讨论分析部分

2.1 截图说明

图 a)横坐标为 eps，表示 DBSCAN 方法中候选点的最大半径；在测试过程中 eps 从 2 取到 vipno 的总数-1，步长为 10；纵坐标是 silhouette 系数。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 eps 取最大值的时候，聚类效果最好。

2.2 相关说明

2.2.1 DBSCAN 算法

与 K-Means 方法相比，能够处理噪音，能够发现任意形状的 Cluster，只需一次扫描，需要密度参数作为种植条件。Sklearn 中 DBSCAN 的两个参数：Eps 和 Minpts 分别表示所选圆的最大半径和一个圆圈内最小的点的个数。其算法是：

1. 任选一个点 p
2. 算出所有和 p 密度可达的点
3. 如果 p 是核心点（以 p 为中心画圈），找出所有与它密度连接的核心点，圆圈融合，一个聚类就形成了
4. 如果 p 是边界点，将 p 划分到 p 所属的圆圈内

其中：

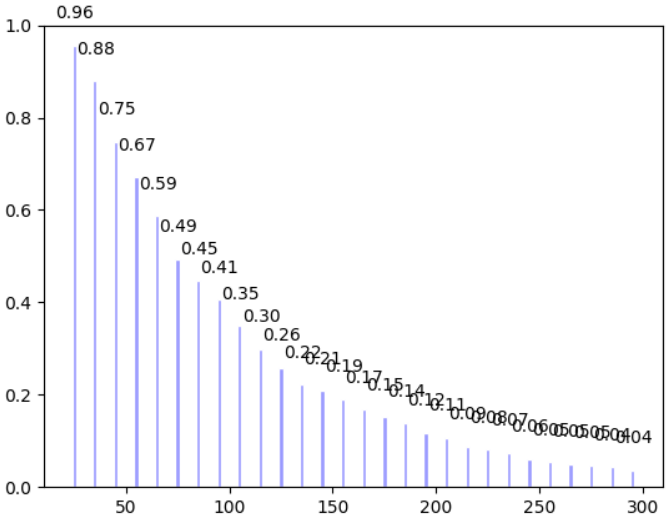
核心点：圆圈个数大于 MinPts

边界点：Eps 内含有个数小于 MinPts 的点，但属于某个核心点的圆内

噪音点：既不是核心点也不是边界点。

2.2.2 效果评估

由于 DBSCAN 可以扫描出噪音点，可以根据下图看出来数据中噪声点所占比：

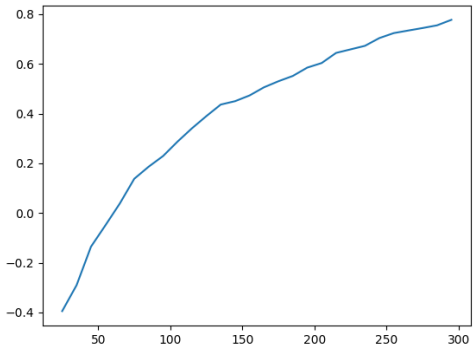


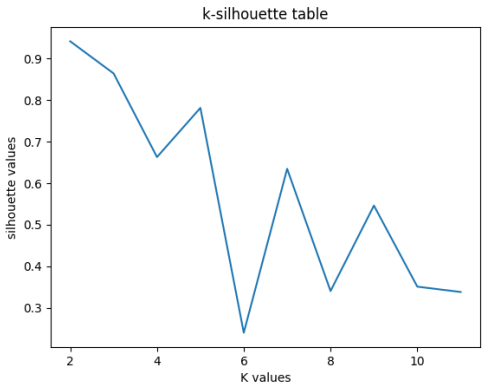
由于噪声数据量较大，整体而言 silhouette 系数不是很高；但随着 eps 的增长，即圆圈的半径增大，数据越可能被分在同一个簇里，其 silhouette 系数就不断增长。

然后看 b 小问的输出结果。经过 lshash 选出任意一个 vipno 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 DBSCAN 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 DBSCAN 聚类结果。结合上述图表得到的信息，可判断绝大部分的数据被标为噪音点，而从桶中随机取到的 vipno 大概率是噪音点中的一个，被分到和它一个桶中的数据也就大概率在 DBSCAN 中被分到了一起。

2.2.3 K-Means 和 DBSCAN 对比

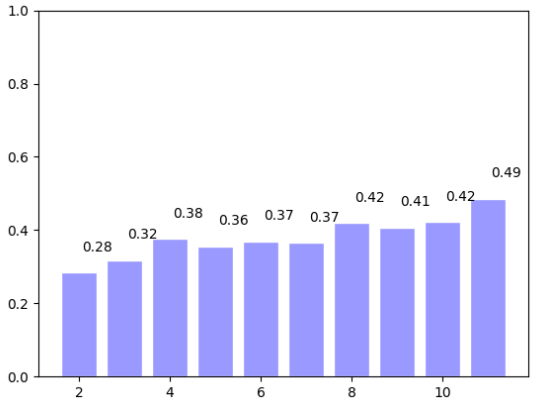
Silhouette 系数的对比：



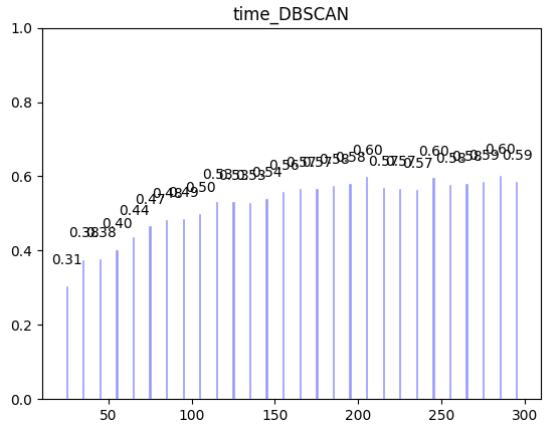


根据对比情况，k-means 聚类效果要高于 DBSCAN，但 DBSCAN 较 k-Means 较为稳定

消耗时间对比：



K-means 时间消耗图



DBSCAN 时间消耗图

由图可看出：随着 eps 的逐渐增大（K 值得增大），DBSCAN（K-Means）的处理时间也在不断增大，但总体而言增幅不大；

与 Kmeans 相比，DBSCAN 在 eps 少得时候对应 K-Means 得 k 值少得时候处理时间长，但随着 eps 增大增幅缓慢；K-Means 得增幅较 DBSCAN 的增幅较大。