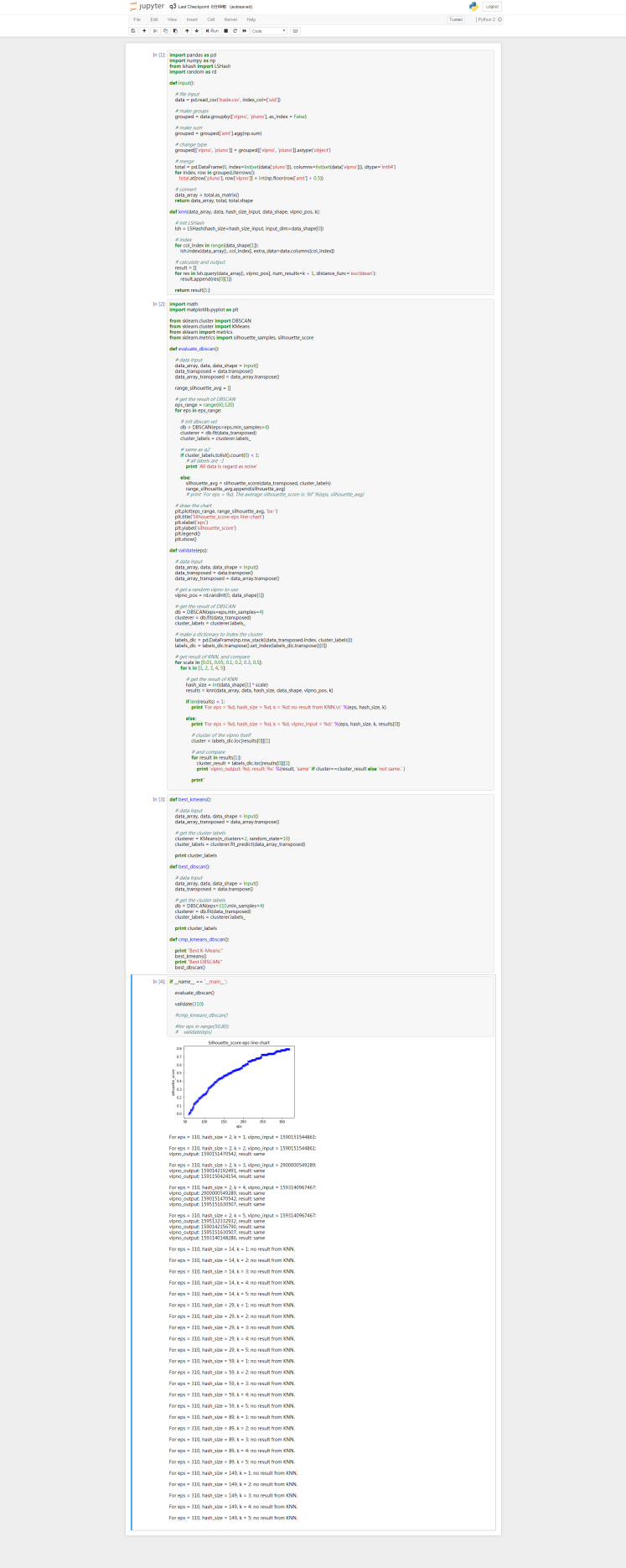
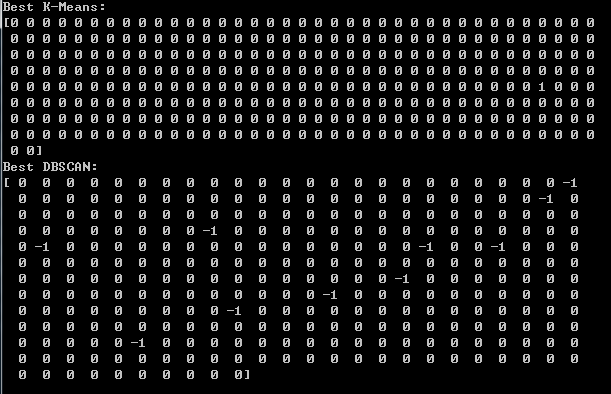
## 代码运行

[点击查看原图](q3.png)



### 讨论分析

1. DBSCAN使用的方法很简单，它任意选择一个没有类别的核心对象作为种子，然后找到所有这个核心对象能够密度可达的样本集合，即为一个聚类簇（至少包含min\_samples个对象）。接着继续选择另一个没有类别的核心对象去寻找密度可达（距离小于eps）的样本集合，这样就得到另一个聚类簇。一直运行到所有核心对象都有类别为止。
2. 本题目中直接使用从q1获得的训练集矩阵以及sklearn的dbscan方法进行聚类。
3. 本题目方案中取min\_samples为2以尝试取得更多聚类簇，在（60,320）的区间内调试eps值计算silhouette\_score。由截图中的变化函数推测，eps值越大，silhouette\_score也不断增大。但是考虑到随着eps值的变大，DBSCAN聚类的耗时也将越长。因此取eps=310，其后silhouette\_score变化较为缓慢，且在这一点有一个可观的跳跃式增长。这表明多加入了一个核心对象，使得聚类的相似度更好。
4. 考虑q2中所知，聚类的作用是过滤离群数据，再参考此时的聚类结果，有个位数个数据被认为是噪声数据被标记为-1，而剩下的数据都被标记为第0簇。故认为聚类有意义。测试调整min\_samples参数从4至2，发现实际聚类结果变化不大，因为噪声数据由于其离群的特征，实际并不能形成聚类。在eps变化的过程中并没有产生第二个有效聚类。
5. 之后对eps=310时产生的DBSCAN聚类，使用KNN的结果进行验证，全部通过。证明这时的DBSCAN聚类有效。
6. 对比最佳取值下的K-Means和DBSCAN的聚类结果，如下图：



可见K-Means实际只去掉了一个噪声数据并标记标签为1，而DBSCAN则去除了个噪声数据，标记为-1。根据两个算法的silhouette\_score函数图可知，最佳取值的K-Means聚类silhouette\_score明显高于DBSCAN，且耗费更少时间做到。但是考虑到DBSCAN下的silhouette\_score是单增的，当eps足够大的情况下，也能做到。而且在未知噪声数据个数的情况下仅去除一个样本，相比起DBSCAN缺少一定的适应能力和动态发现聚类的能力。

### 性能比较

查询相关资料，DBSCAN聚类方法的时间复杂度为O(N^2) 。

进行全部DBSCAN查询所花费的时间为8.04s，减去KNN查询所花费的时间2.30s，实际耗时为5.74s。