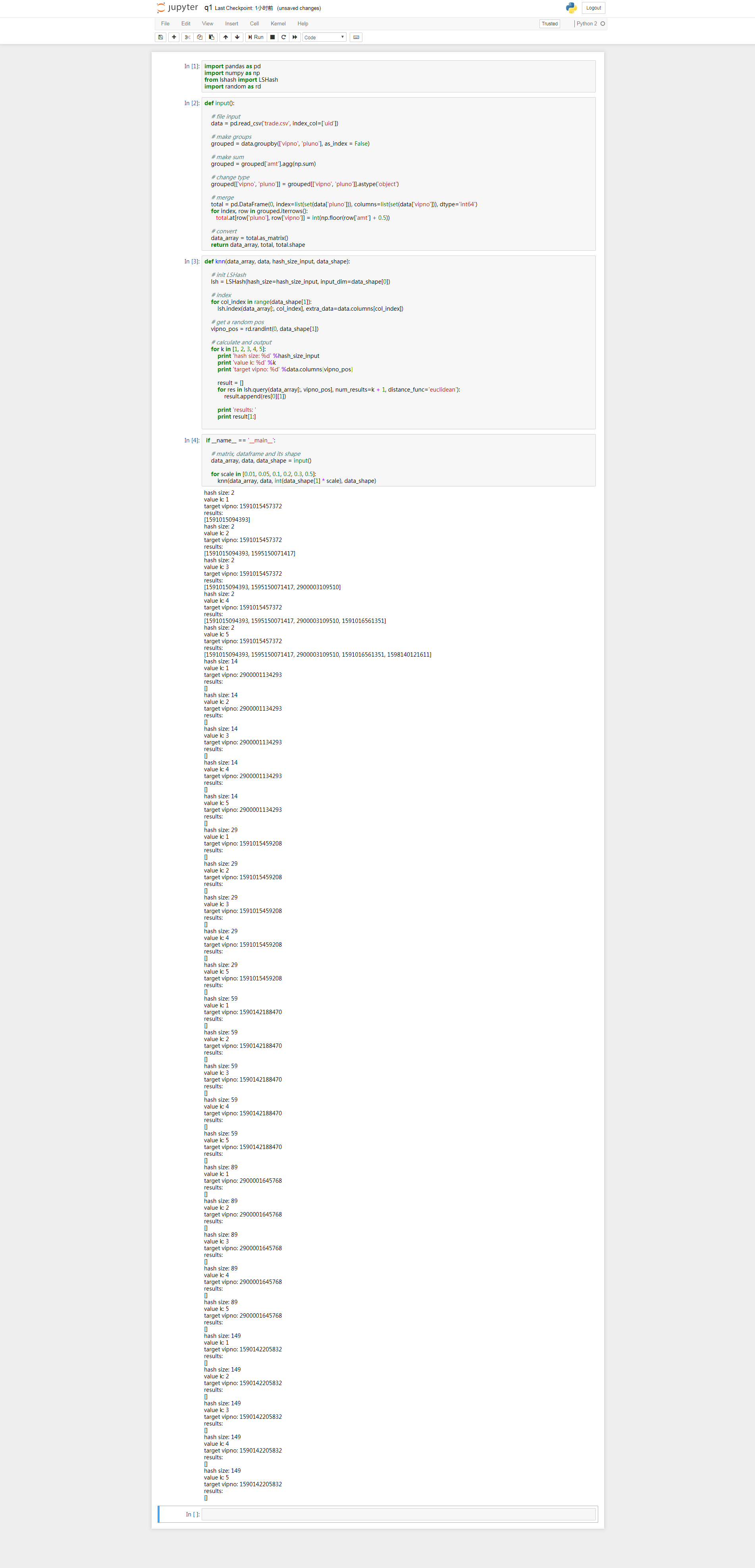
## 代码运行

[点击查看原图](q1.png)



### 讨论分析

1. KNN算法的原理，即对一个需要分析的样本，去训练集中寻找k个距离最相近的数据，若其中大部分的数据属于a类，则将样本归为a类。
2. 使用LSH进行对海量数据建立索引（Hash table）并通过索引来进行近似最近邻查找的过程如下：

a. 离线建立索引

（1）选取满足(d1,d2,p1,p2)-sensitive的LSH hash functions；

（2）根据对查找结果的准确率（即相邻的数据被查找到的概率）确定hash table的个数L，每个table内的hash functions的个数K，以及跟LSH hash function自身有关的参数；

（3）将所有数据经过LSH hash function哈希到相应的桶内，构成了一个或多个hash table；

b. 在线查找

（1）将查询数据经过LSH hash function哈希得到相应的桶号；

（2）将桶号中对应的数据取出；（为了保证查找速度，通常只需要取出前2L个数据即可）；

（3）计算查询数据与这2L个数据之间的相似度或距离，返回最近邻的数据；

1. 本题目中KNN的返回值即为k个与样本距离最近的数据的vipno。又因为样本数据本身也是训练集的一部分，因此这k个数据中包含样本自己。我选择去除样本本身并输出剩下的k-1个数据。且这几个数据按照距离从小到大进行排序。
2. 选择样本进行KNN查询时，随机选择一行样本进行查询，可多次反复运行以防止随机性。
3. 从结果可见，仅有hash\_size为全体数据总数的0.01倍时，KNN有输出。可见当hash\_size过大时，LSHash没有可靠输出。

### 性能比较

距离函数使用欧几里得距离的情况下，全部KNN查询总时间为2.30s。

查询相关资料可知，LSHash的查询时间复杂度为O(logN)或O(1)（取决于采用的索引方法）