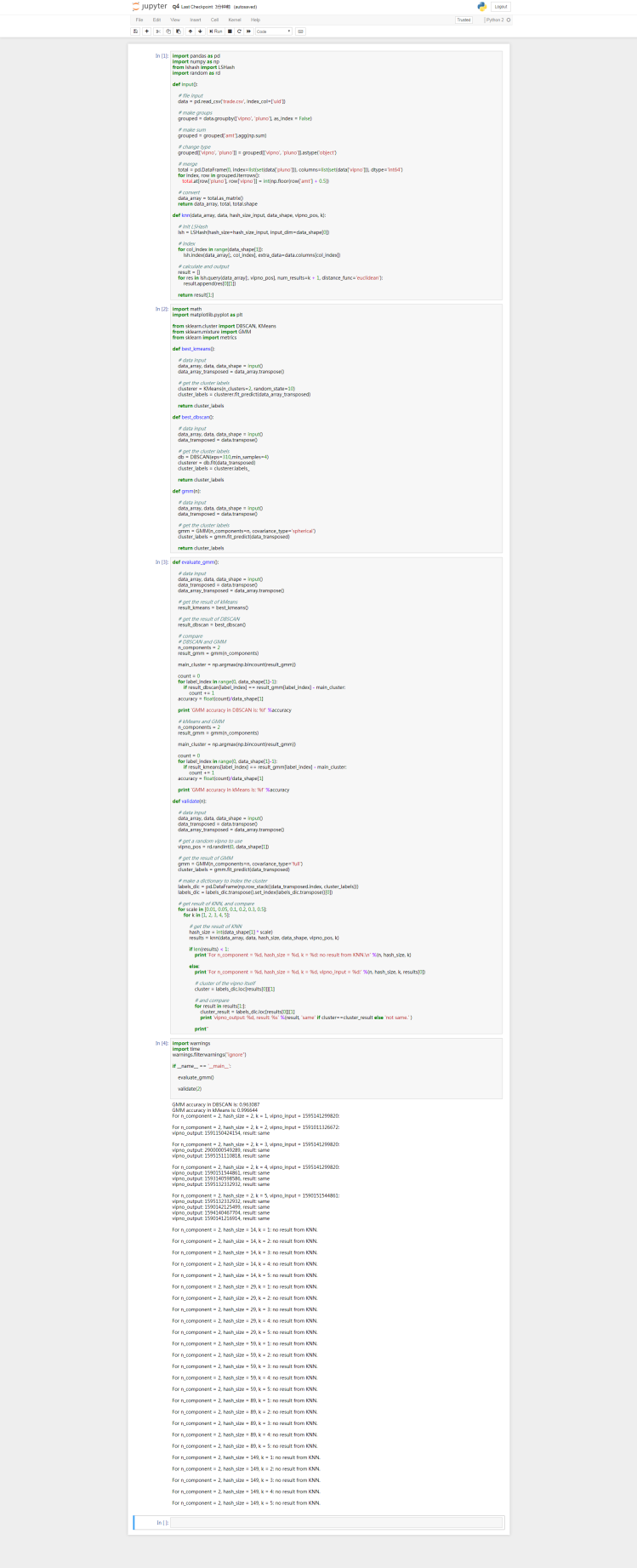
## 代码运行

[点击查看原图](q4.png)



### 讨论分析

1. 那么，利用GMM进行聚类的过程是利用GMM生成数据样本的“逆过程”：给定聚类的簇数N，通过给定的数据集，以某一种 参数估计 的方法，推导出每一个混合成分的参数（即均值向量）、协方差矩阵和权重，每一个多元高斯分布成分即对应于聚类后的一个簇。类似K-Means聚类的过程，但加入了概率的成份。
2. 本题目中直接使用从q1获得的训练集矩阵以及sklearn的gmm方法进行聚类。
3. 由于K-Means产生的簇的数量为2，而DBSCAN产生的簇的数量为1。因此在和K-Means对比时，使用n\_components为2，和DBSCAN对比时应当设置为1。但是若将n\_components设置为1，则产生的GMM聚类簇数将仅有1个，这也就是说，所有的数据都属于一个簇，并没有进行分类，我认为由此计算的GMM聚类并没有意义。因此也取n\_components为2，将噪声数据-1也作为1类数据，来计算聚类准确度。
4. 之后对n\_components=2时产生的GMM聚类，使用KNN产生的结果进行验证，全部通过，证明聚类可靠。

### 性能比较

在设置covariance\_type属性为spherical时，完成全部GMM查询耗时 8.31s，减去KNN查询消耗的2.30s，实际耗时6.01s；

在设置covariance\_type属性为full时，完成全部GMM查询耗时 12.16s，减去KNN查询消耗的2.30s，实际耗时9.86s；

结合之前的KNN、K-Means和DBSCAN聚类方法，绘制性能比较图如下：

注：由于测试数据存在一定偶然性，实际性能可能稍有不同。

可见在同时使用euclidean距离的情况下，KNN算法性能明显优于K-Means和DBSCAN，而后两者大致相同。此外在协方差参数为spherical的情况下，GMM算法性能与K-Means及DBSCAN算法大致相同，但是当协方差参数为full时，GMM耗时明显提高。