数据挖掘实验报告

1，VSM and KNN

2， NBC

3，Clustering with sklearn

姓名：祝瑶佳

学号：201834893

VSM and KNN

项目：为文本建立 **VSM** 并使用 **KNN** 进行文本分类

一、实验方法

1. 掌握文本预处理的方法(提取词干，去除停用词等)。
2. 计算每个文档的词频得到词频矩阵。
3. 计算 TF-IDF 值，计算权重，提取关键词。
4. 得到 VSM。
5. 处理文档得到训练集和测试集。
6. 计算欧式距离。
7. 计算 K 个最近邻并合并。
8. 测试 KNN 分类效果。

二、实验任务

1. 预处理文本数据集，并且得到每个文本的 VSM 表示
2. 实现 KNN 分类器，测试其在 20Newsgroups 上的效果。

三、实验数据

20 Newsgroups

四、实验步骤

1，文本处理

（1）分词

读取文档按空格分词，并且去掉符号。将文档划分成单词，并对单词做一些处理：大写字母变成小写字母，名词复数变单数，去掉停用词，各种时态和形式的动词变成原形，只保留英文词干部分。

（2）划分训练集和测试集

将数据集划分成训练集和测试集，其中训练集占 80%，测试集占 20%。

（3）创建词典

从训练集中读取所有的文档，统计所有的单词及词频，计算 TF-IDF 的值，提取关键词，并创建字典。

（4）得到文本的向量表示 VSM。

2，使用 **KNN** 进行文本分类

将数据集划分为训练集和数据集，计算每一个测试实例到训练集实例的欧式距离，对所有距离进行排序，得到 K 个最近邻。对最近邻进行合并排序，最后测试分类准确度。

五、实验结果

结果如下图：数据集的词频矩阵：

```
d1 = [2, 0, 4, 3, 0, 1, 0, 2]
d2 = [0, 2, 4, 0, 2, 3, 0, 0]
d3 = [4, 0, 1, 3, 0, 1, 0, 1]
d4 = [0, 1, 0, 2, 0, 0, 1, 0]
d5 = [0, 0, 2, 0, 0, 4, 0, 0]
d6 = [1, 1, 0, 2, 0, 1, 1, 3]
d7 = [2, 1, 3, 4, 0, 2, 0, 2]
```

TF-IDF：

```
idf:
[0.24303804868629444, 0.24303804868629444, 0.146128035678238, 0.146128035678238, 0.8450980400142568, 0.066946789630
61322, 0.5440680443502757, 0.24303804868629444]
```

```
>>> distances  # 对应的距离
array([[ 13.37908816,  13.60147051,  13.60147051,  13.60147051,
         13.60147051,  13.6381817 ]])
```

六，实验结论和感想

本实验中，k 值的选取不同，准确率也存在差别，可以多尝试下不同的 K 值，找到使得准确率最高的 K 值。

不管是在 VSM 模型的建立中，还是 KNN 分类中，由于数据量很大，会发生内存溢出的现象。

NBC

项目：使用朴素贝叶斯分类器分类文档

一、实验方法

1. 将文档分成训练集和测试集
2. 将训练集进行分词处理(去除重复词，去掉符号)得到词典
3. 将每个类的所有文档分词得到词典
4. 将训练集得到的词典与每个类的词典进行比较，如果训练词典中的词在类词典中出现，则记录出现次数，没出现则记为 0，得到训练集对应每个类词频向量
5. 统计每个词在每个类出现的次数和此类中所有词出现的次数，计算得到 p(每个词|每个类)
6. 将测试集文档进行分词处理并得到词典。
7. 比较测试集词典和训练集词典，如果有相同的词，则对应词的 p(每个词|每个类)可以从训练集中延用，可以放入测试集概率向量列表中。
8. 将测试集中词典的所有 p(每个词|每个类)相乘再乘以类概率 pclass 得到该文档属于此类的概率，依照此方法计算出每个测试文档属于各个类的概率
9. 比较每个训练文档属于各个类的概率，找到最大的概率，并将文档归于相应的类。

二、实验任务

1，实现朴素贝叶斯分类器，测试其在 20 Newsgroups 数据集上的效果

三、实验数据

20 Newsgroups

四、实验步骤

1，文本处理

（1）分词并创建词典

读取文档按空格分词，并且去掉符号和重复。将文档划分成单词，

（2）划分训练集和测试集

将数据集划分成训练集和测试集，从每个类抽取 80%为训练集，20%为测试集，组成最终的训练集和测试集。

（3）得到训练集对应每个类词频向量

2，训练贝叶斯分类器

（1） 得到训练集的每个词属于每个类的概率向量

3，测试贝叶斯分类器

（1） 根据训练效果得到测试集每个词属于每个类的概率向量

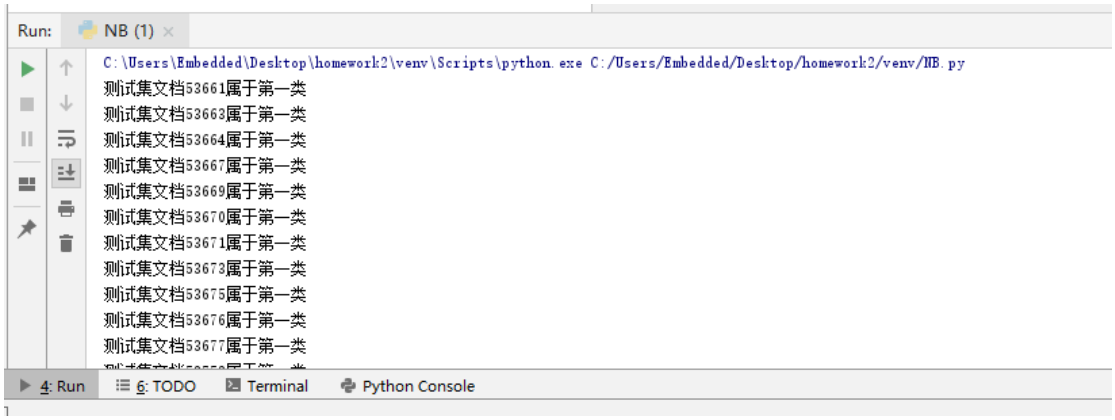
（2） 得到每个文档属于每个类的概率并选择概率最大的，将文档归于此类。

五、实验结果

训练集的概率向量：

```
C:\Users\Embedded\Desktop\homework2\venv\Scripts\python.exe C:/Users/Embedded/Desktop/homework2/venv/NB.py
1.04551186327e-07

2.09102372653e-07

8.36409490613e-07

1.25461423592e-06

1.35916542225e-06
```

分类结果：



六、实验结论和感想

词频向量是统计每个词在每个类的所有文档中出现的次数。

概率向量是统计每个词在每个类的所有文档中出现的概率。

贝叶斯分类器的特点就是简单实用，分类效果也不错。

Clustering with sklearn

项目：测试 sklearn 中聚类算法在 tweets 数据集上的聚类效果。并使用 NMI(Normalized Mutual Information)作为评价指标。

一,实验方法

10. 将文档分词，并记录真实类号和类总数(89 个类)。
11. 计算所得分词的 TF-IDF 的值获得词权重
12. 调用 KMens 方法，返回预测的聚类结果
13. 使用 NMI 评价标准，将真实聚类结果与预测值作比较，得到 NMI 值。

5,调用 Affinity propagation 方法返回聚类结果并计算 NMI 值
6,调用 Mean-shift 方法返回聚类结果并计算 NMI 值
7,调用 Spectral clustering 方法返回聚类结果并计算 NMI 值
8,调用 Ward hierarchical clustering 方法返回聚类结果并计算 NMI 值
9,调用 Agglomerative clustering 方法返回聚类结果并计算 NMI 值
10,调用 DBSCAN 方法返回聚类结果并计算 NMI 值
11,调用 Gaussian mixtures 方法返回聚类结果并计算 NMI 值

二,实验任务

1，测试 sklearn 中聚类算法在 tweets 数据集上的聚类效果。并使用 NMI(Normalized Mutual Information)作为评价指标。

三,实验数据

Tweets.txt

四,实验步骤

1，文本处理

(1) 将文档分词(可根据标点符号分)，并记录真实类号和类总数(89 个类),
(2) 计算所得分词的 TF-IDF 的值获得词权重

2，分别调用 8 种聚类方法(89 个类，与真实类数相符合)

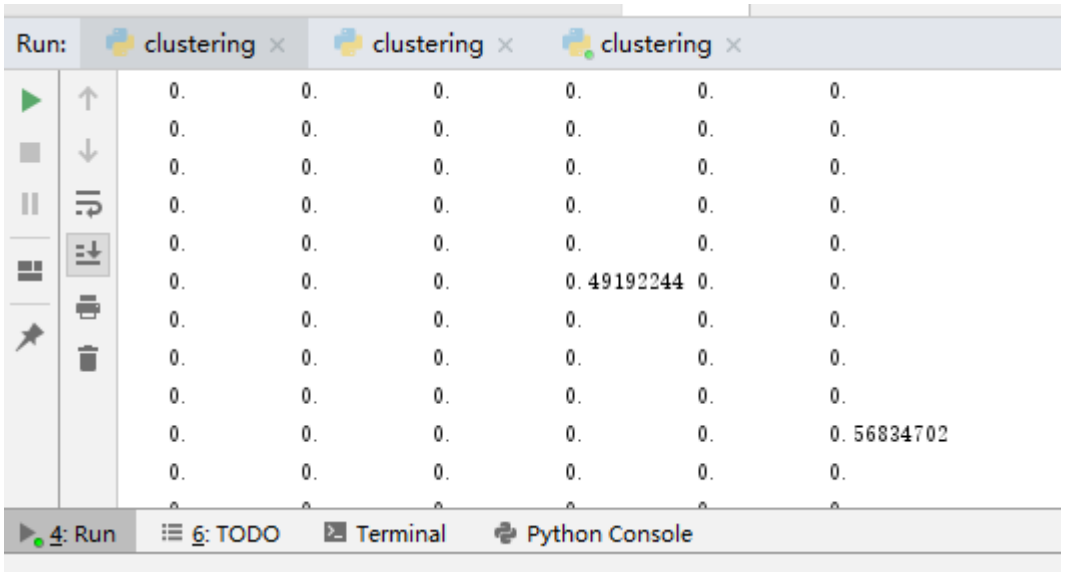
(1)得到预测聚类结果

3，使用 NMI 评价指标分别评价 8 次不同聚类方法所得到的聚类效果

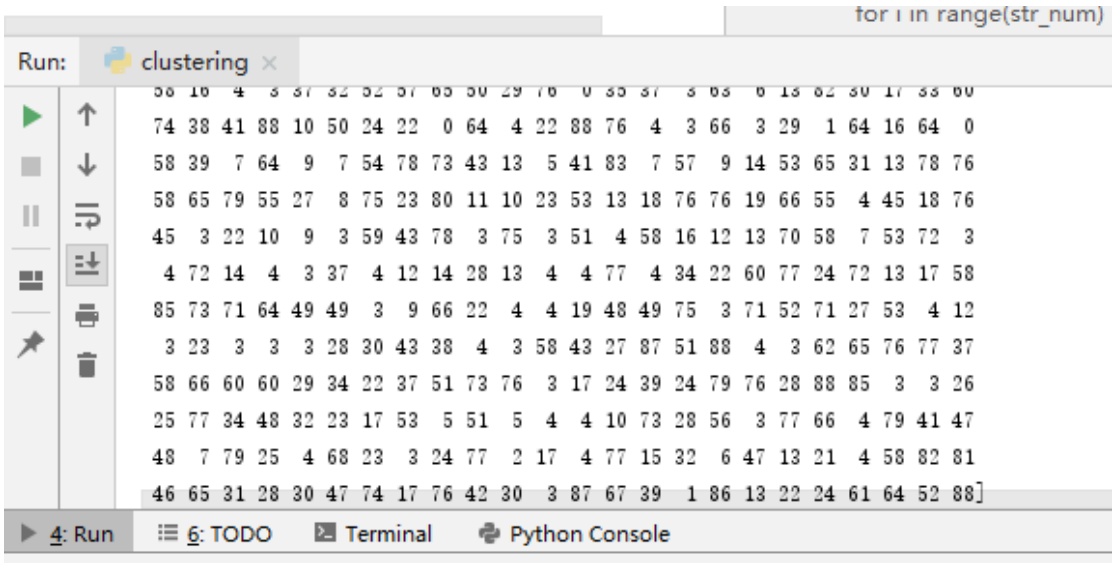
(1)输入预测聚类结果和真实类结果，得到 NMI 的值，NMI 的值越高，说明聚类效果越好。

五,实验结果

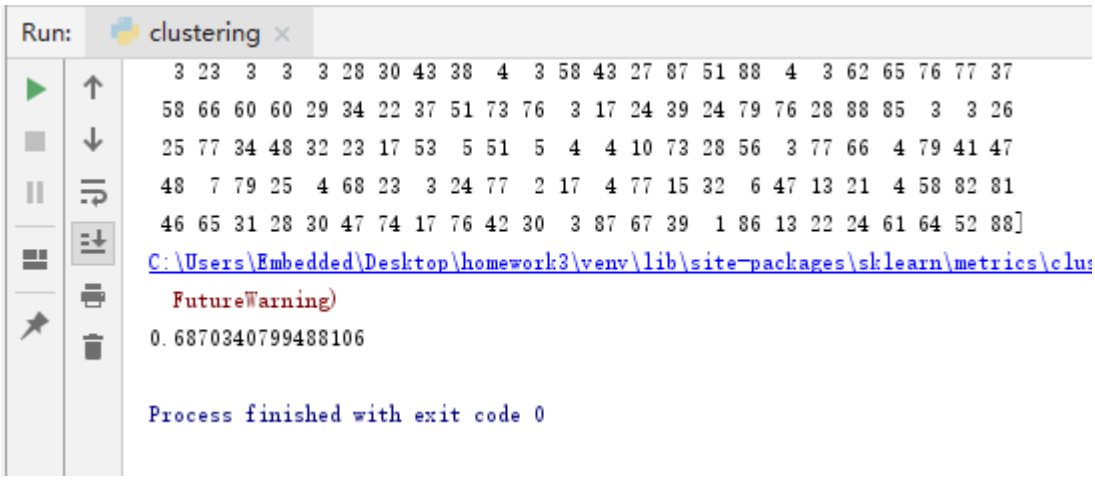
分词的 **TF-IDF** 矩阵：



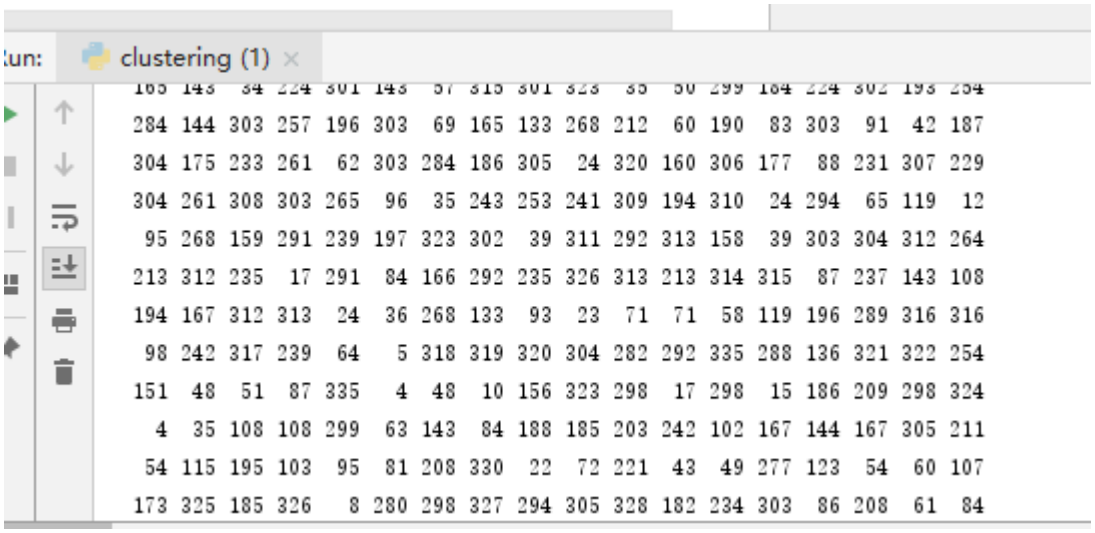
调用 **KMeans** 后的聚类效果：



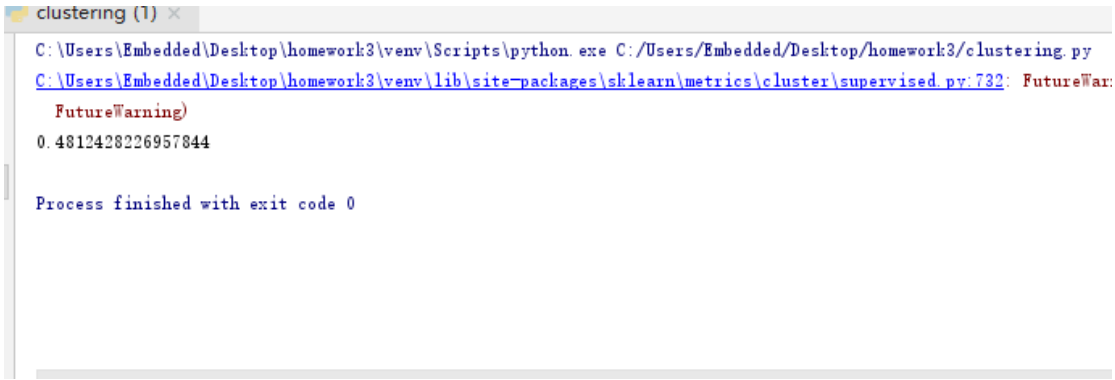
NMI 评价标准的值：



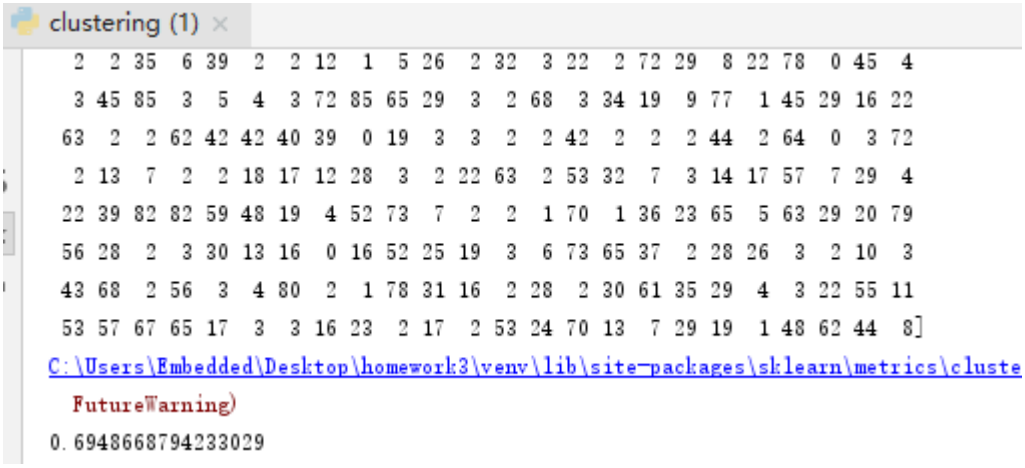
使用 Affinity propagation 的聚类结果



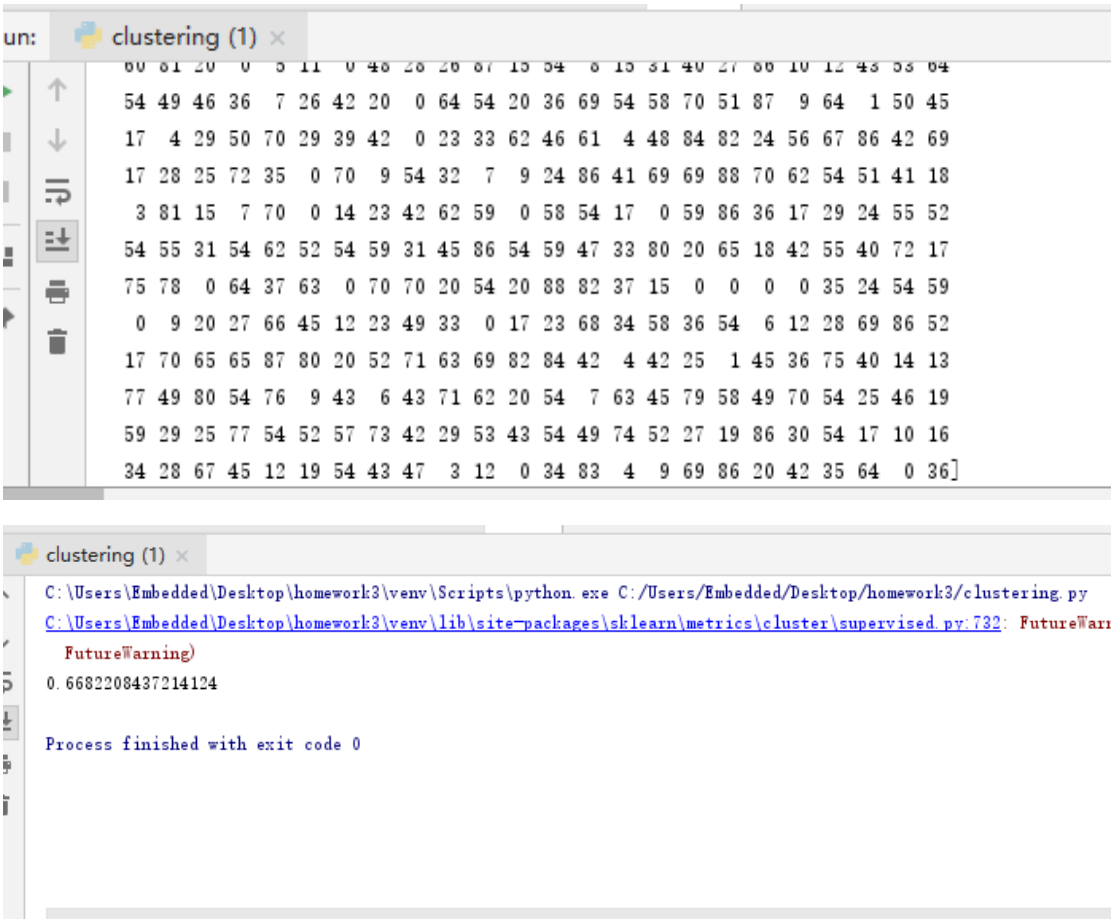
NMI 评价标准的值：



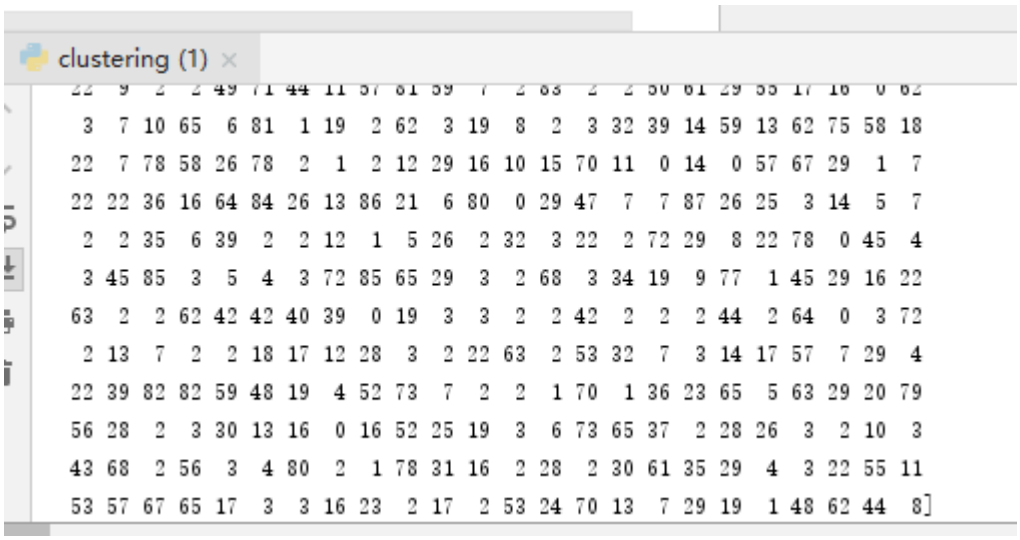
使用 Agglomerative clustering 的聚类结果及 NMI 值

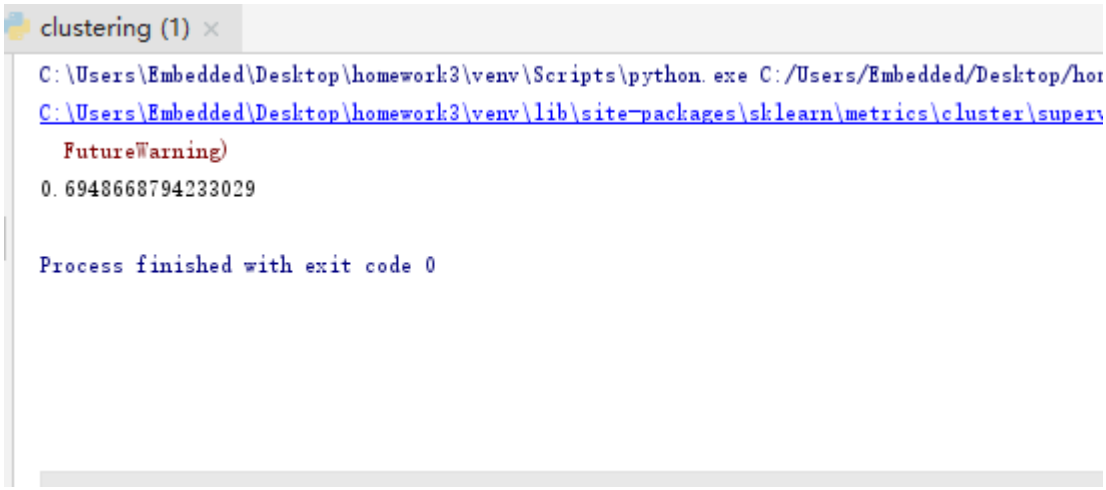


使用 Spectral clustering 的聚类结果及 NMI 值

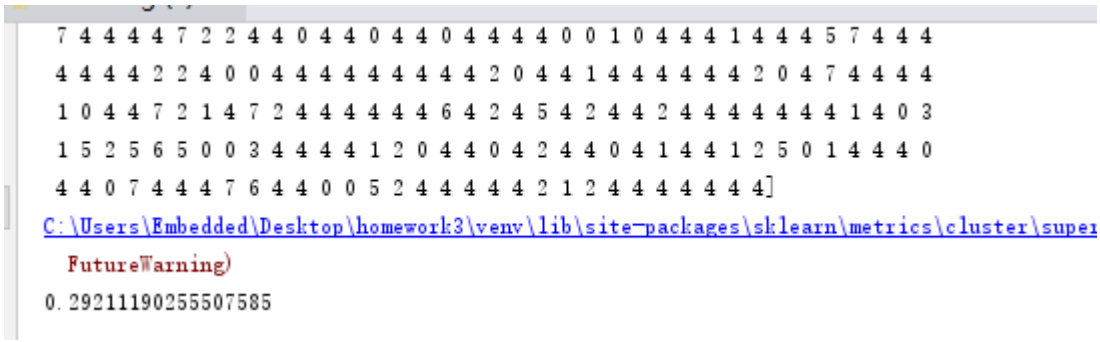


使用 Ward hierarchical clustering 的聚类结果及 NMI 值





使用 Gaussian mixtures 的聚类结果及 NMI 值



六,实验结论和感想

Sklearn 是一个很好的工具，对于一些聚类分类算法，直接调用就可以节省很多时间，但是对于同一个文本，不同的聚类方法效果也不同，有的比较好，有的比较差，而且一些聚类方法比如高斯方法，如果聚类个数增多，NMI 值会提升，但是如果太多就会报错无法聚类。