DM Homework Report

----Sk-Learn 的实现

姓名：曹　瑞

学号：201834856

一、实验要求

（一）预处理文本数据集，使用 sklearn 进行各种 cluster 算法。

（二）统计每种算法的准确率，比较不同的效果。

二、程序设计思路

（一）数据的预处理

1. 数据集的产生

将每一类数据前 75%的文件作为训练数据，后 25%的数据作为完全测试数据集。训练数据用来构建词典以及建立模型。在训练数据中，将每一类数据前 50%的文件作为构建 VSM 的数据，后 25%的数据作为训练数据中的预测试数据。进行词典建立的时候我们预设词典的大小为 500 维，即选取 500 个最具代表力的单词来构建词典。

2. 数据类别过滤

本次实验采用 Stanford CoreNLP 作为分词工具，并通过其对分词的标签功能进行单词类别的过滤。试验中主要考虑动词及其各种时态，名词，地名，书名等与有代表性的词语，其余的单词，例如物主代词，介词，副词等均不列入考虑范围，都需要过滤掉。

（二）算法实现思路

1. 词典的建立

扫描每一类下前75%的文件，对于每个文件中出现的单词，计算其在当前文档中出现的频率TF，以及在其他文档中出现的次数IDF，通过TF-IDF衡量单词的好坏，最终选取500个最具分类能力的单词构建词典。其中IDF算法采用如下公式：

$$IDF(\mathrm{w}) = \log(\frac{D}{1+D_w})$$

其中D表示文档总数，D_w表示文档中出现单词w的文档数。

2. 根据词典描述预测试数据

根据建立出来的词典，先将每一类前50%的每一个文件描述为一个500维的向量。其次将每一类50%-75%的文件进行预测试分类，分别计算每一类每一个属性上的均值与方差。其中概率计算采用如下公式：

$$\mathrm{p}(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i-\mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

其中$\mu_{c,i}$表示均值，$\sigma_{c,i}$表示方差。

三、实验过程

这次实验是使用 sklearn 去实现各种算法，因为之前的实验我都是用的 stanfordCoreNLP 进行词典的建立以及算法的实现，所以这次词典的建立需要重新写程序，遇到了一点小问题，不过在询问同学以后，都得到了解决。

算法实现上，因为以前没接触过 python，所以在文件读取时遇到了一些问题，对于文件的编码有了更深的认识，同时对于 python 语言对格式要求的严格性也有了体会。

四、代码说明

此处仅对 main 函数进行说明，具体的代码详见附录或者源代码

```
if __name__ == '__main__':
    openfile()                                  ##打开词典文件，将文本保存为text，将类别保存到tags
    kmeans(text,tags)
    affinity_propagation(text,tags)
    mean_shift(text,tags)
    spectral_clustering(text,tags)
    agglomerative_clustering(text,tags)
    dbscan(text,tags)
    gaussian_mixtures(text,tags)
```