Data mining:KNN 分类

姓名：宓生润 学号：201834872

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。
3. 20%作为测试数据集，保证测试数据中各个类的文档均匀分布。

数据集：20 Newsgroups dataset（下载地址 http://qwone.com/~jason/20Newsgroups/)

流程:

1. 预处理:

调用模块：os，nltk，textblob

将数据集按照文件夹顺序读取，调用 textblob 对文本进行分词，然后对于进行分词后的文本进行去除 stopwords 的操作，主要是调用了 nltk 中的英文 stopwords 进行处理。然后检查每个分词是否属于英文单词（通过对分词中每个字母进行检查来判断）。之后对于每个分词进行词干提取 (Word 类 ：lemmatize() 方法 对单词进行词形还原，名词找单数，动词找原型。所以需要一次处理名词，一次处理动词)，然后将处理后的文件保存为中间文件以便后续程序进行处理。具体代码见 Preprocessing.py

2. Vector space model:

调用模块：os，collections，random

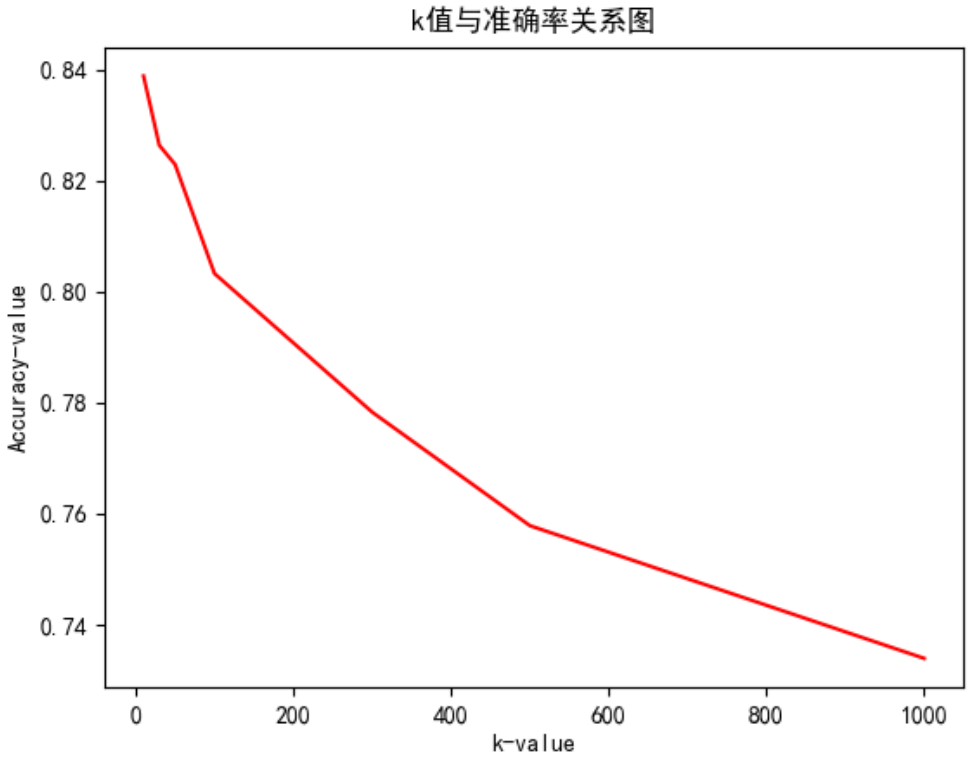
先读取所有文件的文件名，然后利用 random 函数生成百分之二十的随机数，利用这些随机数将数据集分成训练集和测试集，然后根据训练集的地址读取所有训练集的数据生成词典，并统计词频，后将词频小于 9 和大于 10000 的词剔除进行保存，然后利用保存好的词典与训练集和测试集，生成最原始的基于词频的 01 向量。

```
0 subject 16427
1 write 13792
2 would 12761
3 one 12228
4 use 11177
5 get 10743
6 know 8756
7 think 8099
8 like 7779
9 go 7741
10 people 7607
11 doe 7479
12 x 6582
13 time 6479
14 see 5920
15 u 5378
16 work 5254
17 could 5210
18 system 4789
19 new 4786
20 good 4733
21 right 4545
22 need 4520
23 even 4432
24 look 4367
25 problem 4367
26 thing 4339
27 god 4272
```

3. knn 分类及评估:

调用模块：os，gensim，pickle，matplotlib

利用过滤后的词典对训练集进行过滤，将训练集每个文件的分词都保存在列表中，利用 pickle 将中间文件保存以便后面程序调用，调用 gensim 中的 dictionary 对训练集进行处理，然后利用 gensim 对训练集进行 tf-idf 的处理，将处理后的 model 进行保存，然后对测试集进行相同处理，然后对于训练集和测试集进行相似度计算，并将各个文件的 label 保存好以便之后预测后检测是否预测准确，相似度计算完毕后，利用 knn 算法选取不同的 k 值，这将决定着测试集的预测结果，最后将预测后的结果进行统计，最后查看准确率。



关于作业的文件介绍：data 文件夹里保存的均是中间文件以及最后对于 k 值和准确率关系的曲线图，Preprocessing.py 是预处理的代码，VSM.py 是生成向量的代码，knn.py 是 knn 分类以及评估的代码

总结:

1. K 值的选择：对 K 近邻算法的结果会产生重大影响。
K 值较小：就相当于用较小的领域中的训练实例进行预测，“学习”近似误差会减小， K 值的减小就意味着整体模型变得复杂，容易发生过拟合；
K 值较大：就相当于用较大领域中的训练实例进行预测，其优点是可以减少学习的估计误差，但缺点是学习的近似误差会增大。
2. 训练集的规模越大，一般情况下 knn 分类的准确率就越高。
3. 预处理对于最后结果也有着很重要的影响，预处理做的越精细，最后结果都比较满意。
4. 在我设置的 k 值中，准确率最高的 k 值是 10。