一：朴素贝叶斯算法

朴素贝叶斯是基于贝叶斯定理的。即 P(A|B) = P(B|A) P(A) / P(B)。

对于多属性的概率计算，可得

P(A | B1,B2,B3…Bn)=P(B1,B2,B3…Bn|A)*P(A)/P(B1,B2,B3…Bn)

而朴素贝叶斯中的朴素一词的来源就是假设各特征之间相互独立。

即 P(B1,B2,B3…Bn | A)=P(B1|A)* P(B2|A)* P(B3|A)…P(Bn|A)

这一假设使得朴素贝叶斯算法变得简单，但有时会牺牲一定的分类准确率。

二：实现过程

二．1：总体思路

先将数据集按 4：1 的比例分为训练集和测试集。

每一个训练集中的文件夹作为一个类别，对测试集中的每个文件做分析，得到该文件属于哪个类别的结果。

对测试文件中的分类而言，需要对每个类别遍历一遍，求出属于每个类别的概率，取其中概率最大的一个，认为此测试文件属于这个类别。

二．2：计算测试文件属于某类别的概率

求出测试文件中每个单词在该类别中的出现概率，如果该单词从未出现，则表示为类别单词总数分之一。将所有单词概率相乘。

求出每个单词在训练集中出现的概率，如果该单词从未出现，则表示为训练集单词总数分之一。将所有单词概率相乘。

用单词在类别中的出现概率除以单词在训练集中的出现概率，再乘以先验概

率，即为测试文件属于该类别的概率。

其中先验概率表示为类别中所有单词的数量除以训练集中总的单词数量。

二．3：数据结构

数据结构是三个字典。

第一个字典是文件夹字典的字典，即嵌套字典。key 是文件夹路径，value 是文件夹字典。文件夹字典的 key 是 word，value 是 word 在这个文件夹里出现的概率。

另一个是文件夹单词数字典。key 是文件夹路径，value 是文件夹单词数。

最后一个是大字典。key 是单词，value 是单词出现的概率。

三：总结

代码过程比上次 KNN 快了好多，因为 KNN 的许多代码可以直接拿过来用。数据结构也基本类似。但还是有很多差别。KNN 中需要对每个文件建立词典，对每个文件的信息都有关注，但朴素贝叶斯不同。在分类过程中后者只关注类别的信息而不关注类别中具体元素的信息。

最后的正确率为 80%~85%左右。