实验报告

一、实验内容

（1） VSM

构建向量空间一个传统的方法就是：

首先利用分词工具对文本进行分词，然后通过去 stopwords、标点、转化小写等操作创建词典。

最后再通过计算每个单词的 tf 以及 idf 得到每个单词的权重，从而得到每个文件的 VSM.

而在本次作业中，我是用的 sklearn 中的 TfidfVectorizer 直接得到 tf-idf 的特征矩阵，简化了很多工作。

（2） KNN

利用已划分好的数据集：

20news-bydate-test

20news-bydate-train

通过计算两个向量之间的距离，得到训练集与测试集每个文件的相似度，并按由大到小排列，然后设置 K 值，得到前 K 个文件中哪个类型最多，就把测试文件分到哪一类。

K 值不同，正确率也不同：

```
In [9]: runfile('D:/DataMining/HomeWork1/KNN_News/main.py', wdir='D:/DataMining/HomeWork1/KNN_News')
加载训练数据 ...
向量化数据集 ...
计算KNN ...
k = 30, 正确率: 0.702337.
k = 31, 正确率: 0.702735.
k = 32, 正确率: 0.704727.
k = 33, 正确率: 0.704594.
k = 34, 正确率: 0.706054.
k = 35, 正确率: 0.705523.
k = 36, 正确率: 0.707780.
k = 37, 正确率: 0.706585.
k = 38, 正确率: 0.709241.
k = 39, 正确率: 0.707249.
k = 40, 正确率: 0.707913.
k = 41, 正确率: 0.705921.
k = 42, 正确率: 0.706320.
k = 43, 正确率: 0.707913.
k = 44, 正确率: 0.708975.
k = 45, 正确率: 0.709506.
k = 46, 正确率: 0.710967.
k = 47, 正确率: 0.711498.
k = 48, 正确率: 0.710170.
k = 49, 正确率: 0.709904.
k = 50, 正确率: 0.707647.
```

二．实验总结

（1）这次实验中遇到了很多的问题，首先是语言问题，以前接触过一点 python，对它掌握的还不是很熟练，所以先自学了 python，在自学过程中发现了 sklearn 这个库具有很大的便利性，能简化很多繁琐的工作，这一点让我的工作减轻了很多。

（2）其次就是 KNN 编程问题，因为开始并不是很了解他的原理以及细节，通过询问同学以及查找资料，一步一步细化，最终得以理解，可见编程是建立在理解的基础之上的。