数据挖掘实验报告

题目：文本数据的分类与分析

---以新闻文本为本次研究的案例

班级：信息学院软件工程1801

小组成员（满分10分）：

潘冠森（9分），陈剑源（8分），

陈泽熙（8分），姚杰（8分）

时间：2020年12月

1.实验目的

1.1掌握数据预处理的方法，对数据进行预处理

1.2掌握文本建模的方法，对文档进行建模

1.3掌握分类算法的原理，基于监督学习的机器学习方法，训练文本分类器

1.4利用学习的文本分类器，对未知文本进行分类判别

1.5掌握评价分类器性能的评估方法

2.实验要求

目标：新闻文本识别成功率达90%以上

3.实验内容

3.1文本数据的获取

利用爬虫技术从各大新闻网站爬取新闻文本，本次实验使用的

是python的requests库和lxml库。requests库是一个优秀的第三方python爬虫库，使用简单，安全，效率高。lxml 是一种使用 Python 编写的库，可以迅速、灵活地处理 XML 和 HTML。

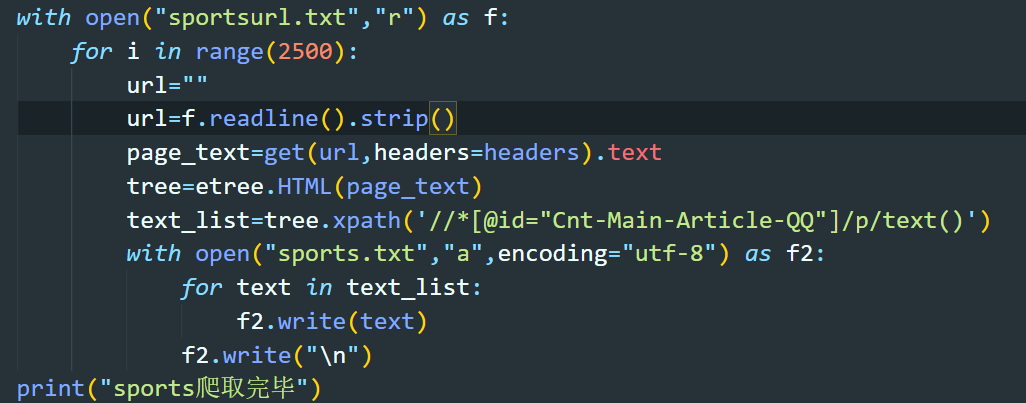
它支持 XML Path Language (XPath) 和 Extensible Stylesheet Language Transformation (XSLT)，并且实现了常见的 ElementTree API。

本次实验为了提高爬取效率，先爬取各个新闻文本的url，存储到本地的文本文件中，再遍历url爬取响应的文本内容到本地文件中存储。

以爬取类别为sports的新闻文本为例，以下为代码：



（爬取url）

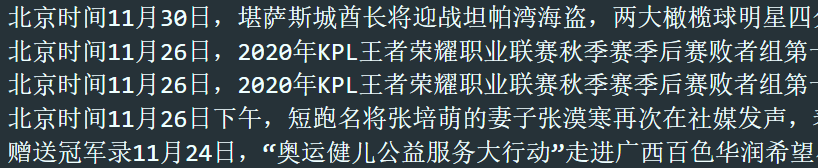


（爬取文本内容）

本次实验共有三个类别：sports，education，technology。

其中sports类共爬取了1635条新闻文本，education类共爬取了1450条新闻文本，technology类共爬取了1386条新闻文本。

以下为sports类部分数据：



（一行为一个新闻文本数据）

3.2数据预处理

上面获得的数据为脏数据，包含了没有意义的标点符号，日期，字，如动词，语气词等，所以我们要先对爬取到的数据进行预处理。数据的成分会极大的影响分类器的构造，进而影响文本识别成功率，因此要对数据进行充分的预处理。

遍历每一条新闻文本，对每条文本作如下处理：

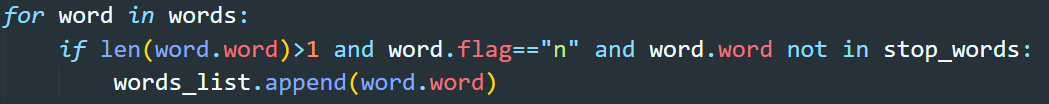
①先把数据中的标点符号和一些无用字符去掉，利用Python正则表达式，使用的是re库，代码如下：



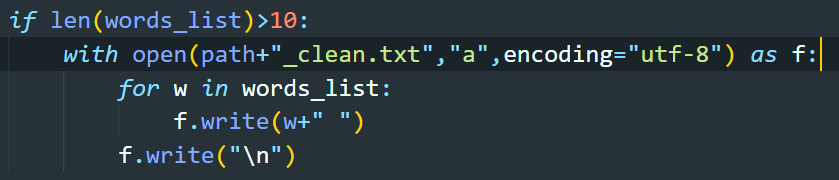
②接下来对文本作分词处理，便于后面的特征提取，分词和词性标注的采用的是python的jieba库中的posseg模块，jieba库是优秀的第三方的中文分词库，分词原理是利用一个中文词库，确定汉字之间的关联概率，汉字间概率大的组成词组，形成分词结果。代码如下：



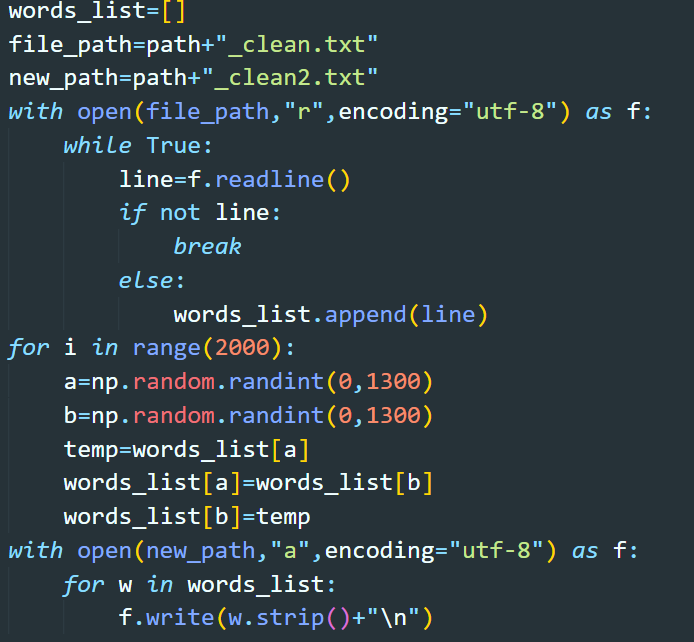
③此时的每一条文本都是一个一个的词组，但并不是所有的词组都是我们需要的，因此对词组进行过滤，只保留名词，过滤掉单字词组，引入停用词表，即把在停用表中存在的词组过滤掉。代码如下：



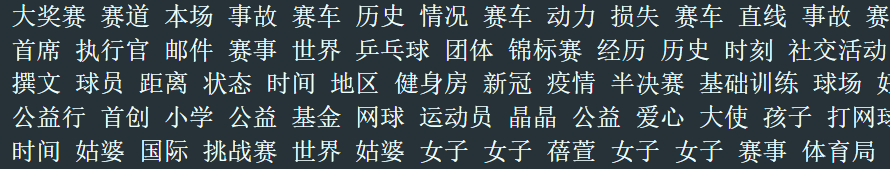
④有些文本词组数过小，不构成文本案例，故去除，将过滤后的干净数据存入本地文件中，代码如下：



⑤因为爬取数据时，相同类别下不同主题的文本数据较为集中，为了避免后面的特征提取时特征词不具有全局代表性，也为了使测试数据具有代表性，在此要充分打乱文本数据的顺序，代码如下：



至此，已得到了我们实验所需要的干净文本数据，以下为sports类部分数据：



将文本数据分为训练集和测试集，数量如下：

|  |  |  |
| --- | --- | --- |
| 类别 | 训练集（单位：条） | 测试集（单位：条） |
| Sports | 1424 | 200 |
| Education | 1248 | 200 |
| Technology | 1127 | 200 |

3.3文本特征提取

本实验采用scikit-learn库，自2007年发布以来，scikit-learn已经成为Python重要的机器学习库了，scikit-learn简称sklearn，支持包括分类，回归，降维和聚类四大机器学习算法。还包括了特征提取，数据处理和模型评估者三大模块。sklearn是Scipy的扩展，建立在Numpy和matplolib库的基础上。利用这几大模块的优势，可以大大的提高机器学习的效率。sklearn拥有着完善的文档，上手容易，具有着丰富的API，在学术界颇受欢迎。sklearn已经封装了大量的机器学习算法。同时sklearn内置了大量数据集，节省了获取和整理数据集的时间。

本步骤将每条文本转化为特征向量，即特征提取。文本分类中最著名的特征提取方法就是向量空间模型（VSM），即将样本转换为向量的形式。为了能实现这种转换，需要做两个工作：确定特征集和提取特征。特征集其实就是词典（词袋），特征权重的计算方式本实验选择TFIDF值。

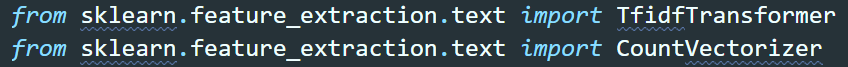
TF值：词频，表示词条在文本d中出现的频率。

IDF值：逆向文本频率，IDF的主要思想是：如果包含词条t的文本越少，IDF越大，则说明词条t具有很好的类别区分能力。某一特定词语的IDF，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取以10为底的对数得到。

TFIDF值：TF值\*IDF值。

特征提取使用的是sklearn库中的CountVectorizer模块和TfidfTransformer模块的组合。本组合会遍历所有文本，提取出特征词向量。

引入代码如下：



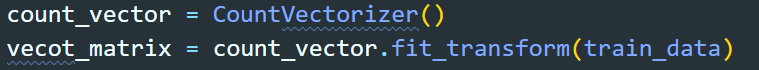
首先遍历所有训练文本文件，将所有文本加载到内存中，把所有文本组合到一个数组中(train\_data),数组每一个元素为一个字符串，内容为一个文本数据，再构造一个数组(tags\_list),每一个元素为一个字符串，内容为train\_data相应下标的文本数据的类型标识。

CountVectorizer：CountVectorizer是属于常见的特征数值计算类，是一个文本特征提取方法。对于每一个训练文本，它只考虑每种词汇在该训练文本中出现的频率。CountVectorizer会将文本中的词语转换为词频矩阵，它通过fit\_transform函数计算各个词语出现的次数。

输入：train\_data数组

输出：词频矩阵（vecot\_matrix）

代码如下：



TfidfTransformer：将词频矩阵转换成TFIDF矩阵。TFIDF矩阵记录的是每一个文本对每一个特征词的TFIDF值。

输入：vecot\_matrix

输出：train\_tfidf

代码如下：



至此，我们得到了训练数据的tfidf矩阵

3.4朴素贝叶斯算法

最为广泛的两种分类模型是[决策树模型](https://baike.baidu.com/item/%E5%86%B3%E7%AD%96%E6%A0%91%E6%A8%A1%E5%9E%8B/17235752)(Decision Tree Model)和朴素贝叶斯模型（Naive Bayesian Model，NBM）。和决策树模型相比，朴素贝叶斯分类器(Naive Bayes Classifier 或 NBC)发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。同时，NBC模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。理论上，NBC模型与其他分类方法相比具有最小的误差率。NBC模型假设属性之间相互独立。

朴素贝叶斯原理：P(B|A)=(P(A|B)P(B))/P(A),如果应用到文本分类中，我们假设有类别集合C={C\_1,C\_2,C\_3},那么文档D属于C\_i的概率就可以使用贝叶斯公式计算：P(C\_i|D)=(P(D|C\_i)P(C\_i))/P(D)。假设文档D的特征集合X有n个特征：X={x\_1,x\_2,…,x\_n},那么P(D|C\_i)的计算公式是：P(D|C\_i)=P(x\_1|C\_i)+P(x\_2|C\_i)+..+P(x\_n|C\_i),

令P(C\_k|D)=max{P(C\_1|D),P(C\_2|D),P(C\_3|D)},那么，我们就判别文档D属于类别C\_k。

本实验采用的是sklearn库中的MultinomialNB模块，引入代码如下：



该模块实现了服从多项式分布数据的朴素贝叶斯算法，也是用于文本分类的经典朴素贝叶斯算法之一。

输入：train\_tfidf(TFIDF矩阵)，tags\_list(类别说明数组)

输出：训练好的文本分类器(clf）

代码如下：



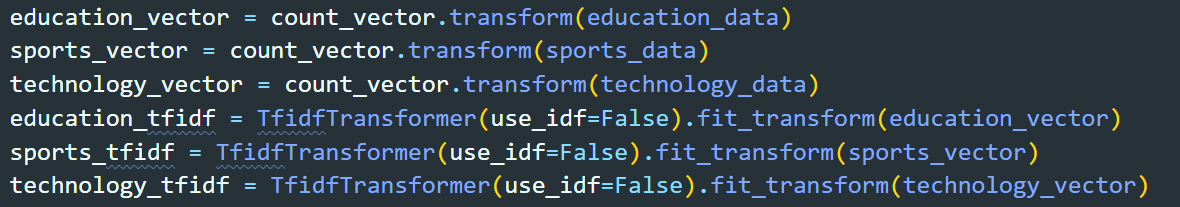
至此，我们得到了训练好的文本分类器

4.实验结果

该步骤将用测试集测试文本分类器（clf）的分类效果。

①先分别读取三个测试集文件，得到三个数组，即education\_data,sports\_data,technology\_data,其中每一个元素为相应类别的一个文本数据。

②使用CountVetctorizer和TfidfTransfromer模块获得各个测试集的TFIDF矩阵，使用步骤如训练集，代码如下：

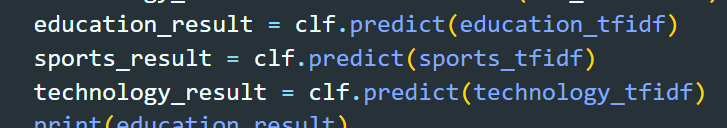


③使用分类器中的predict方法预测各个文本的类别

输入：各个测试集的TFIDF矩阵

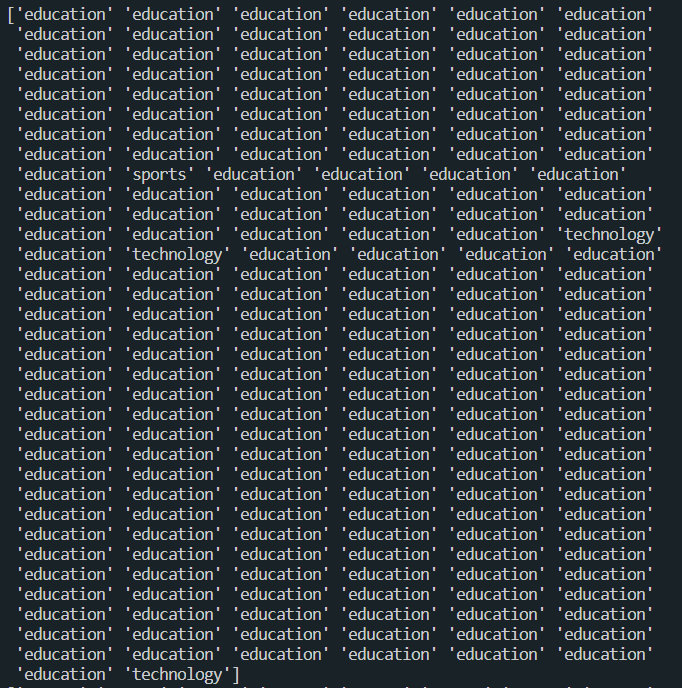
输出：各个测试集的预测结果

代码如下：



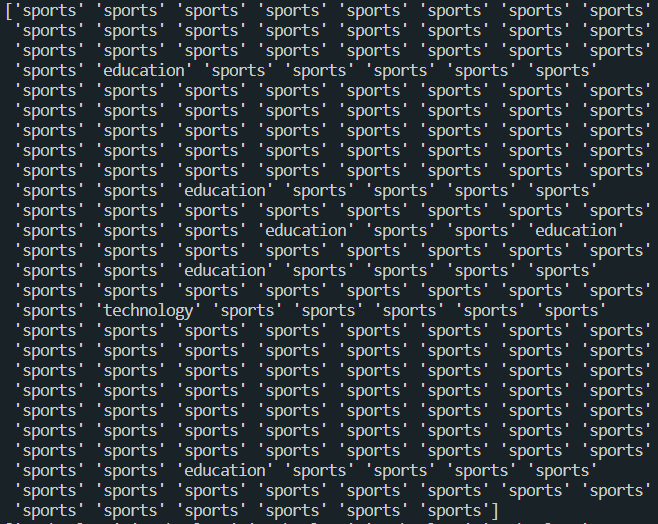
④查看预测结果

education类预测结果：



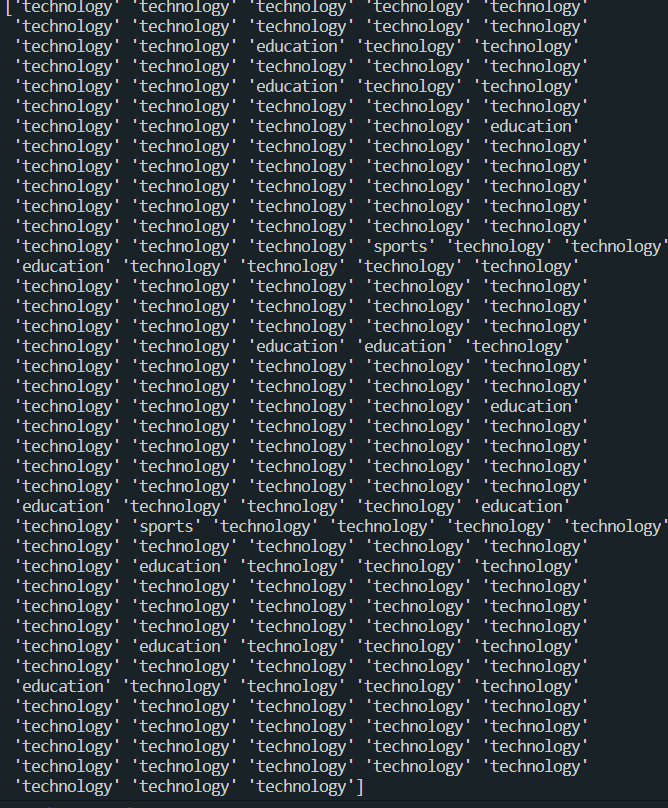


sports类预测结果：





Technology类预测结果：





5.实验总结

本次实验结果成功率均达90%以上。

通过本次实验，我们初步了解了通过机器学习来训练一个分类模型的方法，即构造数据集，通过特征提取得到特征向量，通过机器算法训练模型，测试模型的泛化能力。了解了文本特征提取方法TFIDF方法。了解了朴素贝叶斯的基本原理。

github地址：https://github.com/pgs3413/dataming