

**模式识别大作业**

题 目 垃圾短信分类

学 院 信息科学与工程

专 业 控制科学与工程

组 员 俞天钧(Y30180693)

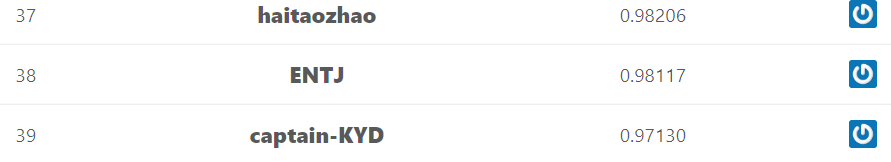
指导教师 赵海涛

**完成日期： 2018 年 11 月7 日**

**模式识别作业报告——垃圾短信分类**

组员：俞天钧

本次大作业我的内容是利用朴素贝叶斯对垃圾短信进行分类，主要工作有数据的采集以及处理，以及使用[TF-IDF](https://deeplearning4j.org/cn/bagofwords-tf-idf)方式来构建词向量，最后使用条件概率来进行分类，实现贝叶斯分类器。通过多次调试以及改进，我最后的结果为：准确率为0.97。



**一、朴素贝叶斯算法所用数据集简介**

本题提供一个数据集, 它包括了5574条英文短信，每条短信内容由几个长短不一的句子组成。每条短信都标注好了是否为垃圾短信，通过该训练集训练出一个分类器，预测短信内容是否为垃圾短信。

[贝叶斯（bayes）分类](http://www.cnblogs.com/leoo2sk/archive/2010/09/17/naive-bayesian-classifier.html)：贝叶斯分类器的分类原理是通过某对象的先验概率，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。

本作业将使用贝叶斯分类器来对垃圾短信进行分类。原始数据集可参见链接：<http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/>

1. **朴素贝叶斯分类器的具体思路及推导**

2.1已知条件概率有：



可以得出贝叶斯分类器的公式为：



其中即垃圾短信以及不是垃圾短信。

既然是用条件概率记忆方法：既然是用条件概率进行分类的，因此绝对是已知某件事情(w)发生了，然后判断这件事情属于哪一类。拿文本分类来说：如果把数据中的短信比喻成为句子，求这个句子是否属于垃圾短信，就是将这个句子的关键词进行提炼，关键词就是判断垃圾短信的关键。w就是这些关键字组成的向量。设=1, =0,即垃圾短信标注为1，正常邮件标注为0如此一来。就是已知w，求*p(1|w)和p(0|w)*，根据贝叶斯分类器的公式，如果求得*p(1|w)>p(0|w)*,那么就认为w属于垃圾短信.

2.2根据题目对贝叶斯分类器进行处理

首先*p(w)*是没有必要参与计算的，根本不影响比较关系。其次 是最容易求得的。比如数据库中共有10条短信，如果有4条是垃圾短信，那么就是*p(1)*=4/10,*p(0)*=6/10;在本文中，我们求的垃圾短信的概率为0.13406317300789664。

比较难处理的是*p(w|1)和p(w|0)*，这里假设每个单词出现是独立的，不会相互影响，即。这样在求联合概率密度过程中，我们就可以将先求每个特征的条件概率再相乘即可。但是我们需要做一些修正。

修正1：如果有一个短信根据辨别后始终没有出现垃圾短信的关键词，那么计算就是0，就会导致也变成0，显然出现以偏概全的现象。为了防止出现概率为0的情况，直接初始化令每个单词在每个类都出现1次。同时将分母(not abusive或是abusive中所有句子的单词出现的次数总和）初始化为2，这是为了使在进行数据训练前，两种类别的概率各占一半。

修正2:由于计算是连乘形式，多个小数连乘由于分母大，分子小，会导致下溢现象出现，导致最终结果的不准确。一种解决办法是对乘积取自然对数。在代数中有 ln(a\*b) = ln(a)+ln(b) ，这样可以避免下溢现象的发生，同时可以简便计算。

* 在本文中，我们求的并不是正真意义上的*p(1|w)和p(0|w)。*但是其原理还属于朴素贝叶斯分类器。

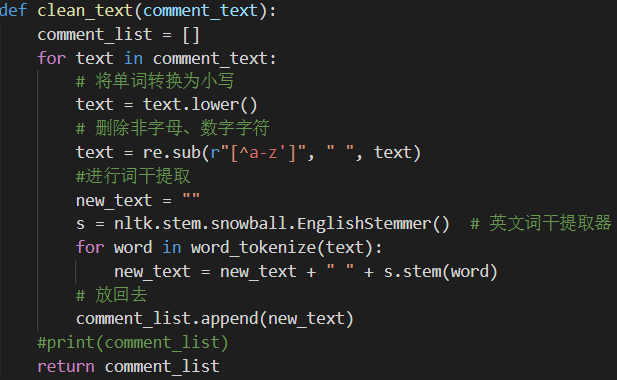
**三、整体解决方案**

**3.1具体步骤分析：**

1. 收集数据：已知给定了数据文件
2. 准备数据：用TF-IDF的方法对数据进行处理，变成向量
3. 分析数据：可视化并观察数据
4. 训练算法：使用建立的trainNB0()函数
5. 测试算法：使用classifyNB()函数
6. 使用算法：将测试得到的结果输出为Submission.csv文件，并计算准确率

**3.2准备数据**

在数据进行大规模的处理之前，我么你首先对数据进行一些“准备工作”，例如将数据都变成小写，这样益于分类；还有删除非字母，数字字符，以及将短信变成若干个词干。



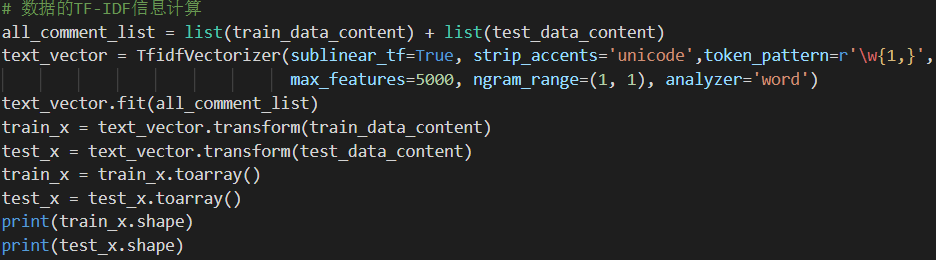
**具体任务：使用了TF-IDF的方法**

TF-IDF（Term Frequency-InversDocument Frequency）是一种常用于信息处理和数据挖掘的加权技术。该技术采用一种统计方法，根据字词的在文本中出现的次数和在整个语料中出现的文档频率来计算一个字词在整个语料中的重要程度。它的优点是能过滤掉一些常见的却无关紧要本的词语，同时保留影响整个文本的重要字词。在本中，重要字词就是垃圾文件的关键词。

Scikit-Learn中TF-IDF权重计算方法主要用到两个类：CountVectorizer和TfidfTransformer。

CountVectorizer类会将文本中的词语转换为词频矩阵，例如矩阵中包含一个元素a[i][j]，它表示j词在i类文本下的词频。它通过fit\_transform函数计算各个词语出现的次数，通过get\_feature\_names()可获取词袋中所有文本的关键字，通过toarray()可看到词频矩阵的结果，TfidfTransformer用于统计vectorizer中每个词语的TF-IDF值。

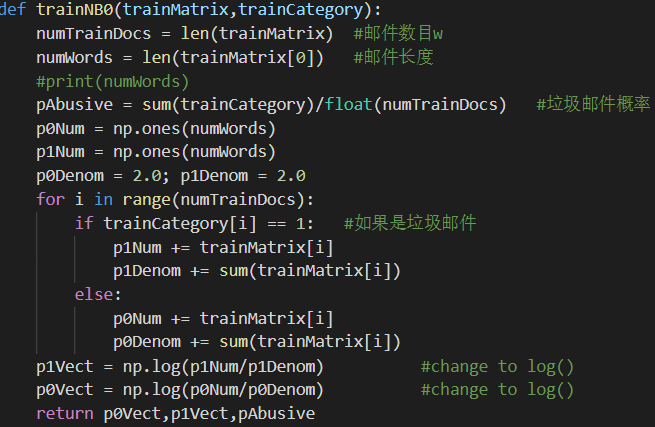
这里我们使用CountVectorizer即可：



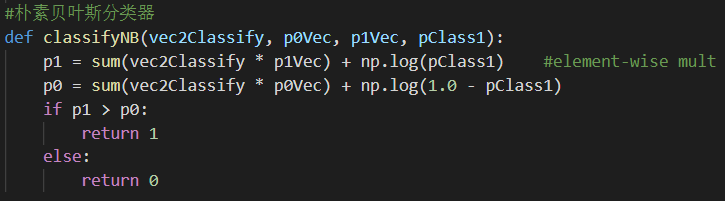
我们一共设有5000个特征值，即：['a', 'aa', 'aah', 'aaniy', 'aaooooright', 'aathi', 'ab', 'abbey', 'abdomen', 'abeg', 'abel', 'aberdeen', 'abi', 'abil', 'abiola', 'abj', 'abl', 'abnorm', 'about', 'abouta', 'abov', 'abroad', 'absenc', 'absolut', 'abstract', 'abt', 'abta', 'aburo', 'abus', 'ac', 'academ', 'acc', 'accent', 'accentur', 'accept', ……]。同时在输出每个句子中包含特征词的个数，所以每一个短信都会变成一个词频向量。

**3.3训练算法及程序实现**

通过这个代码，我们求出垃圾短信的概率以及非垃圾短信的概率，并且依靠对*p(1|w)和p(0|w)*进行求解，同时我们使用上方2.2中的两种修订的方法进行改进，完整代码如下：

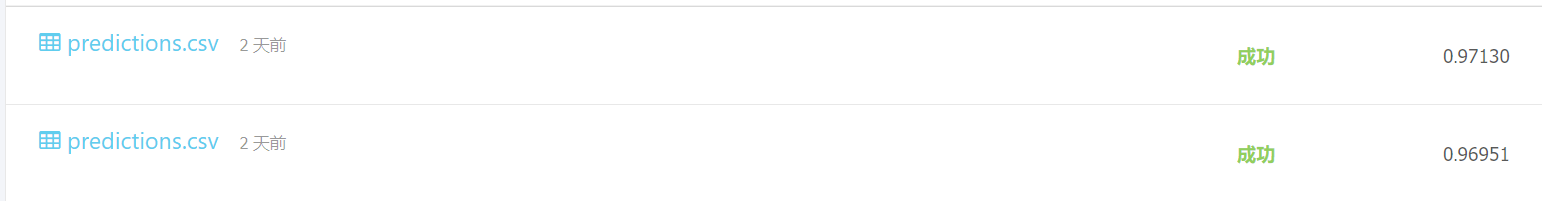


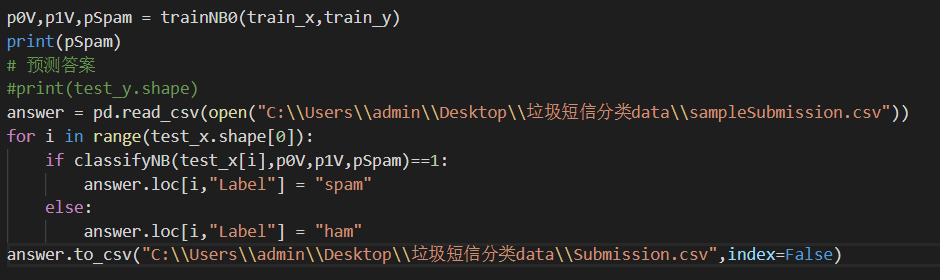
**3.4测试分类器**

根据”如果求得*p(1|w)>p(0|w)*,那么就认为w属于垃圾短信。”进行编写而得。

**3.5调试及预测结果**

通过运行贝叶斯分类器，我们最终得到了对测试样本的分类预测，同时通过对数据预处理方法的微调，将准确率略微调高，并且检验得到准确率为0.97。





**四、作业总结**

本次实验，我通过参考机器学习实战这本数，并且通过题目本身所给的提示，最终使用朴素贝叶斯分类器对实验样本进行分类，虽然不能得到100%的正确率，其主要原因是算法简单，以及优化不够的原因。比如我没有使用PCA算法对向量进行降维。通过这次实验，我发现虽然算法看似简单，但是实现过程中需要根据题目情况进行很多的改变，如果以后还有时间，我或许可以用其他的方法来修正分类器，让其更加出色。

**附：文件说明**

本次附件一共包含有：

1 大作业报告；

2 最终的Python实现程序源码：logistics.py

3 训练集与测试集文件