# 《机器学习实战》

第四章 朴素贝叶斯

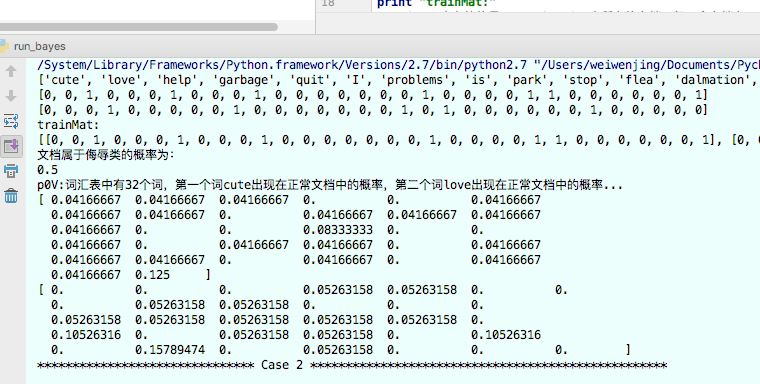
4.5用python进行文本分类

例1：

相应的：



最终结果:



trainMat矩阵内容:

[[0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1],

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0],

[1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1],

[0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1],

[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0]]

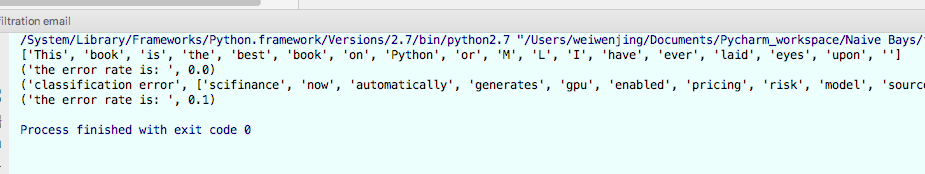
例2：根据现实情况

*#将所有词的出现数初始化为1，将分母初始化为2——避免概率为0  
#为了避免下溢，对乘积取自然对数*



实例：使用朴素贝叶斯过滤垃圾邮件

运行结果：



不太明白训练函数的意义？返回几个概率。。。

实例：适用朴素贝叶斯分类器从个人广告中获取区域倾向

介绍这个实例的背景之前，首先介绍两个东西。

一个是**[RSS（Really Simple Syndication，简易信息聚合）](https://zh.wikipedia.org/wiki/RSS" \t "_blank)**。它是一种**消息来源格式规范**，用以聚合经常发布更新数据的网站，例如博客文章、新闻、音频或视频的网摘。RSS文件包含全文或是节录的文字，再加上发布者所订阅之网摘数据和授权的元数据。把新闻标题、摘要（Feed）、内容按照用户的要求，“送”到用户的桌面就是RSS的目的。

也就是说，RSS是一种**格式**，这种格式的文件以比较规范的形式（当然肉眼读起来还是比较费劲）存储了网页上的信息，例如会包括一条条帖子的内容、帖子的作者、发布的时间等等。可以利用各种各样的**RSS阅读器**去抓取、解析里面的信息。

感觉这个过程有一点类似于爬虫，只不过RSS文件一般都是网站官方制作而成的，目的就是方便用户订阅、获取网站里面的信息（如新闻、帖子什么的），用RSS阅读器爬取、解析RSS文件是受到网站官方欢迎的。**爬虫**的话，就有点“非法”的意味了。按我的理解，爬虫是把网页以HTML文档的方式截取下来，然后根据自己的需求，解析HTML文档，把里面自己想要的内容给“抠”出来（顺便推荐一个HTML解析器，[beautifulsoup](https://www.crummy.com/software/BeautifulSoup/bs4/doc/index.zh.html" \t "_blank)）。爬虫行为一般是不受网站官方欢迎的，不然你以为网站里的**验证码**是干什么的？就是用来防网络数据采集程序的。

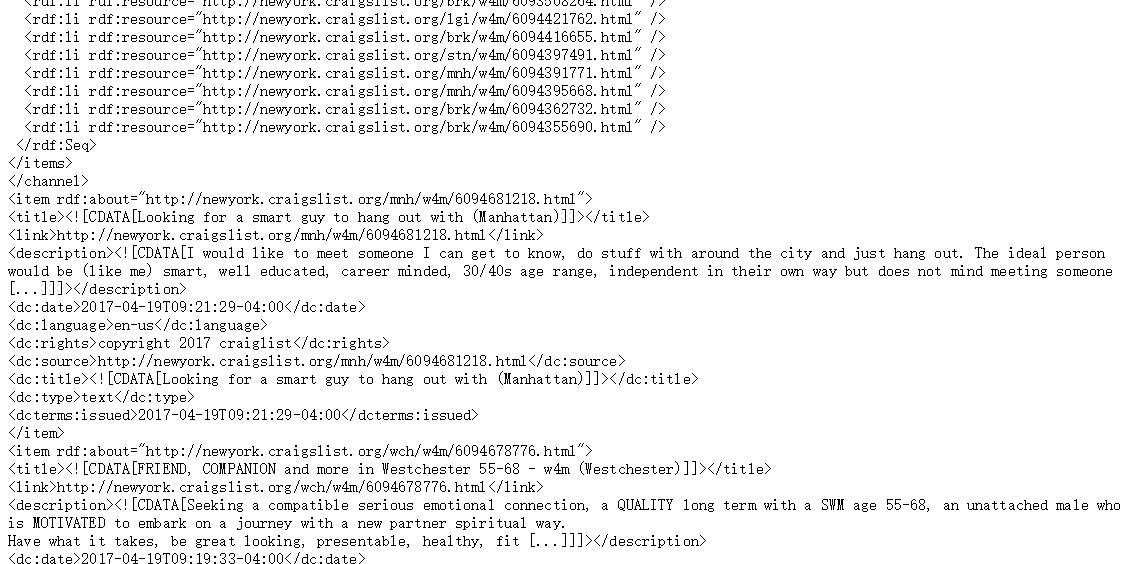
好了又扯远了。说到RSS阅读器，这里推荐的是谷歌家的**[feedparser](https://pypi.python.org/pypi/feedparser" \t "_blank)**，是个Python包，一装就能用。

第二个要介绍的是我们要获取数据的网站，美国的[Craigslist（克雷格列表）](https://newyork.craigslist.org/" \t "_blank)。按百度的介绍，它是个“巨大无比的网上分类广告加BBS的组合，虽然看上去颇为乏味，可是却是美国人最喜欢的网站之一。有人在这里卖掉自己的旧车，有人在这里租到中意的房子，有人在这里找到工作，还有人在这里找到女朋友。”

这是个...不得了的网站啊【手动滑稽】。我随便翻了一下，发现挺多寂寞男女聊骚的帖子...噗

**然而重点是，这个网站提供RSS文件。**

比如我们随便找一个主题：[women seeking men](https://newyork.craigslist.org/search/w4m" \t "_blank)【再次手动滑稽】，在网址后面加上?format=rss，就可以得到该网页的RSS文件了，看一眼：



构建的这个分类器的作用是给出一条帖子，判断（猜测）它是来自那个地区的。New York是1，San Francisco是0.

这份代码思路和上面那个垃圾邮件的代码没什么区别，都是先把文本拆成词，然后构建训练数据集、测试数据及，最后训练、测试，看错误率。

只不过，这里面，调用了**calcMostFreq**函数把全文（所有帖子加在一起）中找出最高频的30个单词，并从词汇表里去掉了。去掉的原因，是因为这些词一般都是“虚词”，只是起一些辅助性的作用，比如中文里的“在、的、是、和、了”之类的。**它们对表征帖子的来源区域并没有很大的帮助。**但更好的方法是人工把这些词列出来，然后在文档里面找到并且去掉，而不是统计高频词。而且这样从经验上看会使错误率降低。这里有一份英文的[**停用词表**](http://www.ranks.nl/stopwords)。

运行结果：

