

**模式识别大作业（二）**

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 何雨旻（Y30180644）、罗伟杰（Y30180662）

李楠希（Y30180656）、马遥（Y30180663）

指导教师 赵海涛

**完成日期： 2018 年 12月16日**

1垃圾短信分类问题介绍

1.1 垃圾短信

随着移动互联科技的高速发展，信息技术在不断改变着我们的生活，让我们的生活更方便，其中移动通信技术己经在我们生活起到至关重要的作用，与我们每个人人息息相关。短信作为移动通信的一个基础业务，由于简单方便而且廉价，在很久以前就己成为人们交流沟通的一种方式，得到人们的认同，成为人们生活中不可缺少的一部分，以短信拜年为例，2015年春节期间，全国短信发送量达到203.9亿条，2016年春节期间，全国短信发送量达到139.6亿条。从上述数据我们可以看出，随着科技的发展，短信可能会被其它社交方式代替，如微信、QQ等，但是我们发现它们之间区别并不大，所以研究垃圾短信的分类问题对研究其它社交方式垃圾信息自动识别具有同样意义。

垃圾短信是指未经用户同意向用户发送的用户不愿意收到的短信息，或用户不能根据自己的意愿拒绝接收的短信息，主要包含以下属性：（一）未经用户同意向用户发送的商业类、广告类等短信息；（二）其他违反[行业自律](https://baike.baidu.com/item/%E8%A1%8C%E4%B8%9A%E8%87%AA%E5%BE%8B/4655858)性规范的短信息。

信息技术改变着我们的生活，我们享受着它带给我们的方便快捷，与此同时我们也常遭受它带给我们的各种困扰，在现在这个信息安全缺乏保障的时代，我们时常会收到一些我们所谓的垃圾短信，不少不法分子利用短信进行勒索、诈骗、传播不实消息和黄色信息，手机用户需要花费大量精力去对这样的垃圾短信进行处理，对于那些诈骗短信一些辨识能力差的人群可能很容易上当受骗，遭受不必要的财产损失，对社会危害不小。垃圾短信危害手机用户权利的同时也会给移动通信带来负面的影响，运营商需要消耗资源对其过滤，垃圾短信己经成为社会的公害。

1.2 垃圾短信分类

如何将垃圾短信和正常短信区分开来，减少垃圾短信的危害，提高正常短信的利用率，实现短信的自动分类具有十分重要的意义。对于垃圾短信的过滤不仅有利于维护手机用户的权益，同时也为移动通信的发展提供一个健康有序的环境。目前短信自动分类技术大都是基于短信文本，也可以说短信分类是一种文本挖掘。根据文本挖掘的原理，将文本挖掘的原理运用到垃圾短信的分类过滤，可以实现短信的自动分类，从而对垃圾短信进行识别过滤，减少垃圾短信给人们生活带来的各种困扰，让短信更好的方便人们的生活。

国外的文本分类研究开始于20世纪50年代末，早期文本分类需要人为的来定义分类规则，这种方法实现起来比较麻烦，同时有太多的人为主观因素在里面，分类效果并不是太好。后来，H.P.Luhn提出了词频统计的想法，为现有文本分类技术奠定了基础。同时期，人们提出不少分类效果比较好文本分类模型，如1960年马龙和库恩提出了概率引索模型，还有索尔顿提出的向量空间模型，这些模型在现在做文本分类还时常被用到。

九十年代，随着各种文本的出现和机器学习的快速发展，人们将文本分为已知类别的训练文本和未知类别的待测文本。首先，使用已知类别的文本对分类模型进行训练，建立判别规则或分类器，最后对那些未知类别的待测文本进行分类预测，在大量实践中它的分类效果很好，这使它在文＾分类中得到广泛应用。如1992年，刘易斯在论文《信息检索中的陈述与学习》比较统的详细的介绍了文本分类系统实现的方法。后来，人们又在提高分类预测的效果方面做了很多工作，包括特征选择、特征降维和分类技术的选择等。如杨益民对特征选择进行了比较，讨论选择文档频率、信息增益和互信息等的分类效果比较，1995年，Corinna Cortes和Vapnik等人于提出了机器学习中经典的分类方法——支持向量机，再后来人们使用非线性的支持向量机来对文本进行分类，后来人们又提出了一些组合方法，并且取得较好的分类效果。

国外常用的文本分类算法有Logistic、支持向量机、朴素贝叶斯、决策树、免近邻、组合方法等方法。与国外相比，我们国内对文本分类研究起步较晚，国内最先提出文本分类研究的是侯汉清教授，1981年，他对国外的文本分类研究的现状、使用的分类技术和取得的成果做了详细的介绍，后来越来越多人们开始接触研究，并取得了不错的成绩。如1995年，吴军等人率先研制出关于汉语语料的自动分类系统；19978年，张月杰、姚天顺等人研制出了第一个关于中文新闻文本分类的自动分类系统；1999年，邹涛、王继成等开发的中文技术文本分类系统。除此之外在国内还有很多人为致力于解决中文文本分类中的各种问题默默的进行着研究，他们希望通过对不同问题的研究找到合适的方法使得文本分类的效果达到最好。如针对于不同语种的文本分类模型，隐含语义在中文文本中的处理技术，最大熵模型等。

1.3 本文主要工作

本文主要工作是首先对数据数据预处理方法、Logistic回归的原理进行简单的阐述。然后，对短信的文本信息进行数据预处理并使用基于逻辑回归的文本分类技术，建立相应的分类器，以实现对短信自动分类，将正常短信和垃圾短信准确的区分开，以实现垃圾短信的过滤。其中本文使用的短信文本是lintcode中垃圾短信分类训练集的英文文本，主要包括垃圾短信和正常短信两大类。

2 数据预处理

要对短信文本进行文本挖掘，我们首先要对数据进行清洗，即对获取到的文本数据进行预处理，因本次主要针对英文文本进行研究，所以我们以英文文本为例，在本次大致将文本处理流程分三个步骤：Normalization和 Stemming，即标准化和词干提取。

2.1 标准化（Normalization）

得到纯文本后，第一步通常要做就是Normalization。在英文中，所有句子第一个单词的首字母一般是大写，有的单词也会全部字母都大写用于表示强调和区分风格，这样更易于人类理解表达的意思，但是从计算机的角度来说是没法区别“Car”、“car”、“CAR”是否是一个意思的，因此我们一般把文本中所有字母都转换为小写或大写(通常意义上是小写)，每个词用一个唯一的词来表示。

例如在下面的代码中，字符串文本调用lower()函数就可以将所有字母转换为小写形式。

|  |
| --- |
| pre\_str = 'I Love My Family'  after\_str = pre\_str.lower()  print(after\_str) |

输出结果为：

|  |
| --- |
| i love my family |

经过大小写统一化之后，我们还需要做的是清除文本中的句号、问号、感叹号等特殊字符，并且保留字母表中的字母和数字。文档分类和聚类等应用中若要将所有文本文档作为一个整体，那么正则表达式这个方法特别有效。用正则匹配小写“a”到“z”以及大写“A”到“Z”以及数字“0”到“9”的范围之外的所有字符并用空格代替。这个方法无需指定所有标点符号。当然，也可以采用其他正则表达式。

例如在下面的代码中，使用re模块的sub正则匹配所有非a-z，A-Z，0-9的字母，并将其替换为空格。

|  |
| --- |
| import re  text = 'the first time you see the second renaissance it may look boring.look at it at least and definitely watch part 2.it will??'  text = re.sub(r'[^a-zA-Z0-9]', " ", text)  print(text) |

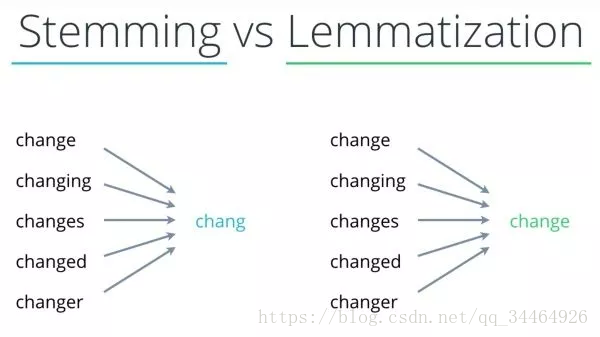
输出结果：

|  |
| --- |
| the first time you see the second renaissance it may look boring look at it at least and definitely watchpart 2 it will |

2.2 词干提取（Stemming）

在语言形态学和信息检索里，词干提取是去除词缀得到词根的过程─—得到单词最一般的写法。对于一个词的形态词根，词干并不需要完全相同；相关的词映射到同一个词干一般能得到满意的结果，即使该词干不是词的有效根。从1968年开始在计算机科学领域出现了词干提取的相应算法。很多搜索引擎在处理词汇时，对同义词采用相同的词干作为查询拓展，该过程叫做归并。

一个面向英语的词干提取器，例如，要识别字符串“cats”、“catlike”和“catty”是基于词根“cat”；“stemmer”、“stemming”和“stemmed”是基于词根“stem”。 很多词干提取是基于Porter词干提取算法写出来的。Martin Porter在2000年发布了一个基于[该算法](http://tartarus.org/~martin/PorterStemmer/)的官方版本的免费应用软件。他在自己的工作上进行延伸，建立了一个Snowball算法，是编写词干提取算法的框架，并实现了一个改良的英文词干提取器可以同时提取一些其他语言，本文所应用的就是Snowball算法。

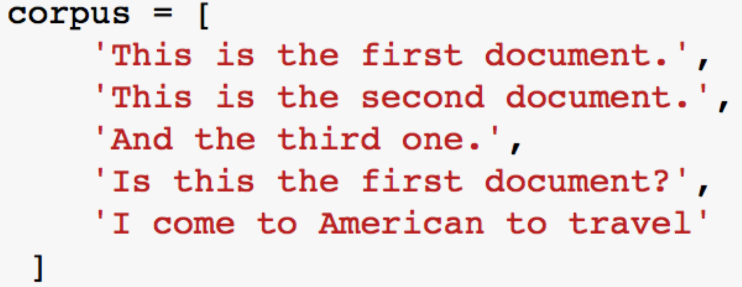


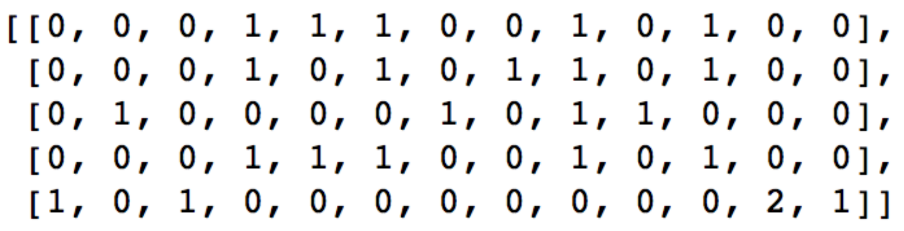
例如，本文主要研究的是英语文本，则可以用如下代码：

|  |
| --- |
| >>> import nltk.stem  >>> s = nltk.stem.SnowballStemmer('english')  >>> s.stem('imaging')  u'imag'  >>> |

2.3 TF-IDF特征向量转换

TF-IDF（termfrequency–inverse document frequency）是一种用于资讯检索与资讯探勘的常用加权技术。TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF-IDF加权的各种形式常被搜寻引擎应用，作为文件与用户查询之间相关程度的度量或评级。除了TF-IDF以外，因特网上的搜寻引擎还会使用基于连结分析的评级方法，以确定文件在搜寻结果中出现的顺序。

一般情况下，将文本分词并向量化后，就可以得到词汇表中每个词在文本中形成的词向量，以下面的5个短文本为例做了词频统计：

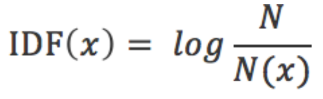
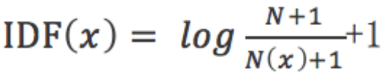
不考虑停用词，处理之后得到的词向量如下：

如果直接将统计词频后的13维特征做为文本分类的输入，会发现有一些问题。比如第五个文本，我们发现"come","American"和“Travel”各出现1次，而“to“出现了两次。似乎看起来这个文本与”to“这个特征更关系紧密。但是实际上”to“是一个非常普遍的词，几乎所有的文本都会用到，因此虽然它的词频为2，但是重要性却比词频为1的"China"和“Travel”要低的多。如果向量化特征仅仅用词频表示就无法反应这一点，TF-IDF可以反映这一点。下面给出TF与IDF的定义：

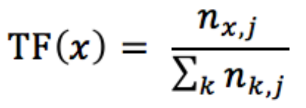
在一份给定的文件里，词频(term frequency, TF)指的是某一个给定的词语在该文件中出现的次数。这个数字通常会被归一化（分子一般小于分母区别于IDF），以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词频，而不管该词语重要与否。）

逆向文件频率 (inverse document frequency, IDF) 是一个词语普遍重要性的度量。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。

而TF-IDF的主要思想是如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。TF-IDF实际上是：TF \* IDF，TF词频(Term Frequency)，IDF反文档频率。TF表示词条在文档d中出现的频率。上面是从定性上说明的IDF的作用，那么如何对一个词的IDF进行定量分析呢？这里直接给出一个词x的IDF的基本公式如下:

其中，N代表语料库中文本的总数，而N(x)代表语料库中包含词x的文本总数。在一些特殊的情况下上面的公式会有一些小的问题，比如某一个生僻词在语料库中没有，则分母变为0，IDF就没有意义了，所以常用的IDF需要做一些平滑，使得语料库中没有出现的词也可以得到一个合适的IDF值，平滑的方法有很多种，最常见的IDF平滑公式之一是：

进而可以推导计算某一个词的TF－IDF值：

其中TF(x)是指词x在当前文本的词频。这个数字是对词数的归一化，以防止它偏向长的文件，（同一个词语在长文件里可能会比短文件有更高的词数，而不管该词语重要与否）。对于某一特定文件里的词语x来说，它的重要性可表示为：

以上式子中分子为x词在第j个文件中出现的次数，而分母是在第j个文件中所有字词的出现词数之和。

在此思想下，我们通过词干提取，得到了分词后的文档序列后，即可使用HashingTF的transform()方法把句子转化为成特征向量。可以看到，分词序列被变换成一个稀疏特征向量，其中每个单词都被散列成了一个不同的索引值，特征向量在某一维度上的值即该词汇在文档中出现的次数。最后，使用IDF来对单纯的词频特征向量进行修正，使其更能体现不同词汇对文本的区别能力，IDF是一个Estimator，调用fit()方法并将词频向量传入，即产生一个IDFModel。很显然，IDFModel是一个Transformer，调用它的transform()方法，即可得到每一个单词对应的TF-IDF度量值。最后可以发现，特征向量将被其在语料库中出现的总次数进行了修正，通过TF-IDF得到的特征向量，在接下来可以被应用到相关的机器学习方法中。

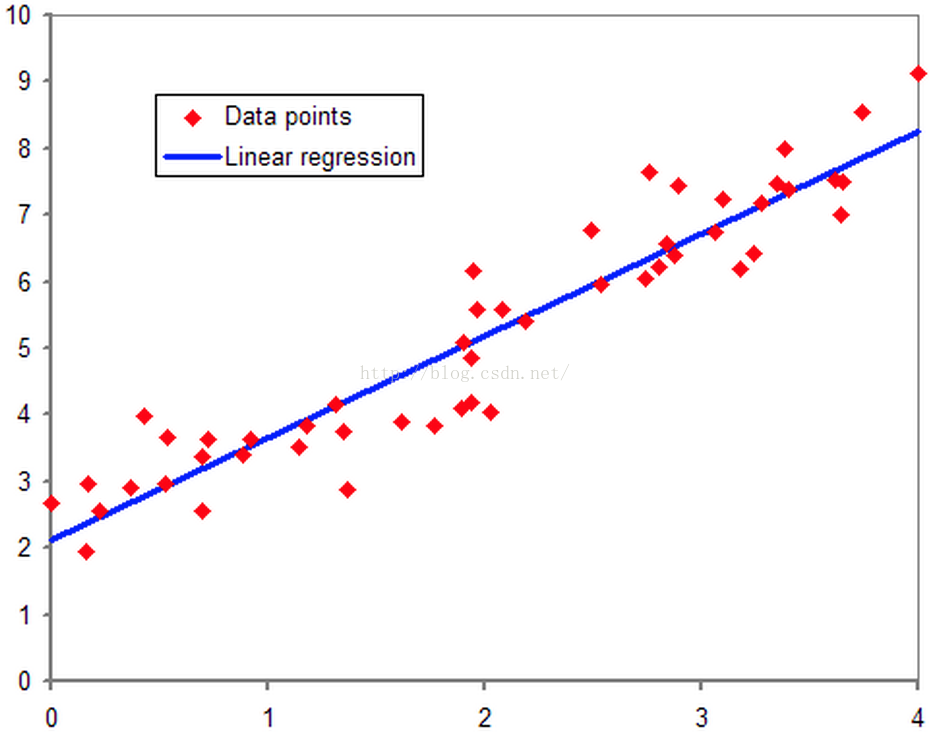
3 建模方法

**3.1 logistics回归**

logistics回归是应用非常广泛的一个分类机器学习算法，它将数据拟合到一个logit函数(或者叫做logistic函数)中，从而能够完成对事件发生的概率进行预测。在垃圾短信分类中，我们将logistics回归用作对垃圾短信的分类，因为logistics回归的特点说明了此种方法对二分类问题的分类效果很好。

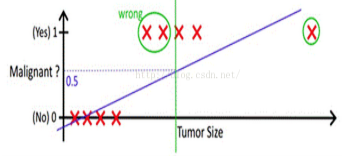
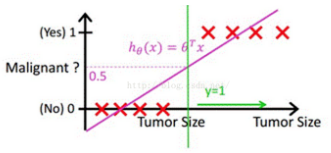
**3.1.1 logistics回归的由来**

我们都学过线性回归，这是一种对多维空间中存在的样本点，用特征的线性组合去拟合空间中点的分布和轨迹的方法。如下图：



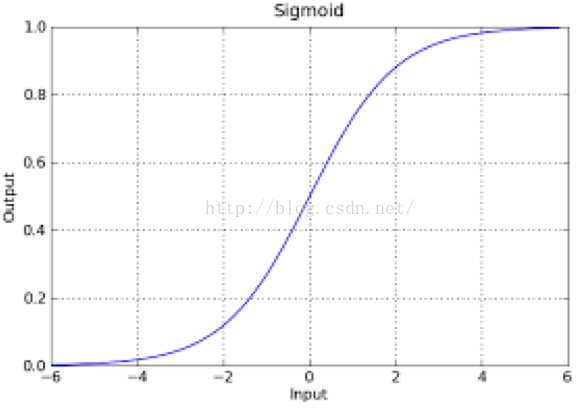
线性回归虽然可以对连续值结果进行预测，但是现实生活中更常见的问题是分类问题。最简单的情况是：是与否的二分类问题。比如说医生需要判断病人是否生病，银行要判断一个人的信用程度是否达到可以给他发信用卡的程度等等。

遇到这类问题的时候我们最直接的想法是，既然能够用线性回归预测出连续值结果，那根据结果设定一个阈值限定一下是不是就可以对这类问题进行求解呢？对于数据非常标准的时候也可以解决，但是现实中我们需要用来学习的数据并不都是那么精准的。



如上图所示，右图阈值作用下可以很好地预测数据，但是左图中在阈值作用下还是有数据不能被回归模型描述出来，那么这个阈值就是失效的。这只是一个很简单的例子，现实中得到的数据会比这个复杂许多倍，因此线性回归模型就不能用了，实现不了我们想要的分类效果。当线性回归的结果输出是一个连续值，而值的范围是无法限定的，人们寻求把这个结果值映射为可以帮助我们判断的结果，因此逻辑回归就诞生了。为了能更好的做回归，人们假设输出结果是 (0,1) 的一个概率值，这个问题就很清楚了。于是找到了数学上的一个简单函数了， sigmoid函数，这个函数的特性表示如下：

sigmoid函数的图如下：



从函数图上可以看出，函数y=g(z)在z=0的时候取值为1/2，而随着z逐渐变小，函数值趋于0，z逐渐变大的同时函数值逐渐趋于1，而这正是一个概率的范围。

**3.1.2 logistics回归模型**

针对二分类问题（），当所选的判别函数，即可认为，，即可认为。考虑正态分布的判别函数在的情况。取后验概率作为判别函数：

针对二分类问题，只有和两类，且有，假设，可以得到以下简化的判别函数：

取d维多元正态分布密度函数作为概率密度函数：

经过计算可以得到：

其中：

于是有：

再使用最大似然估计的想法对问题进行简化，可以得到样本的概率密度函数的参数的似然函数，为了方便问题的分析以及利用梯度上升算法进行参数的求解，使用似然函数的对数函数（Log Likelihood，LL），即目标函数:

其中：，从而得到。目标函数称为logistics回归的损失函数，也就是说logistics回归就是要求，使得最大。

**3.1.3 梯度上升法求解Logistic回归**

经过前面的讨论我们已经得到了logistics回归的损失函数，接下来就是如何找到合适的向量来使损失函数最大。我们首先通过求解的一阶导数和二阶导数来判断其性质。求导过程如下所示：

一阶导数为：

二阶导数：

其中：

从而可以发现（半正定），因此logistics回归的损失函数是连续、可微、二次可微的凹曲线（开口向下）。根据梯度上升的思路，我们只要计算的梯度，再根据下面的递推公式进行梯度上升就可以得到的最优解：

梯度上升公式为：

其中为学习率，在垃圾短信分类中是会随着每次迭代从而减小，这样可以减小随机梯度上升的回归系数波动问题，同时也同时比普通梯度上升收敛更快，加入常数项避免alpha变成0。

3.2 随机梯度上升算法

如果使用传统的梯度上升算法，每次更新权值都会遍历整个数据集，如果处理100个左右的数据集的耗时尚可，但是在本次实验中，数据量稍大，所以使用的是随机梯度上升算法。

随机梯度上升算法即一次只用一个样本点来更新回归系数，由于可以在新样本到来时队分类器进行增量式的更新，因而随机梯度上升算法是一个在线学习算法。与“在线学习”相对应，一次处理所有数据被称为“批处理”。

对应的，随机梯度上升算法可以写成伪代码形式：

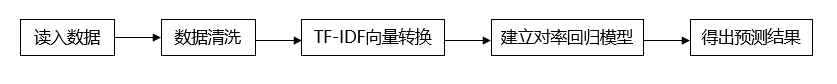
|  |
| --- |
| 所有回归系数初始化为1；  随机选择样本计算梯度；  使用alpha×gradient更新回归系数；  删除已计算的样本向量  返回回归系数值； |

可以看到，随机梯度上升算法和原始梯度上升算法之间其实是十分相似的。但是随机梯度上升法中处理的都是数值，没有矩阵之间的计算，减少了计算机的工作量。此外，因为随机梯度上升法在迭代过程中会出现局部波动现象，所以对随机梯度上升方法又进行了一些改进。将其中学习步长改为下式所示,其中i表示样本点所处位置的下标，j表示迭代的次数。可以看到，学习步长并不是严格单调下降的。并且用于更新权值的向量的选择方法也改为随机选择。

4 仿真实验

4.1 实验策略

根据第二章所述的用来处理垃圾短信的数据预处理方法以及第三章所述的建模方法，我们可以得出如下图所示的解决垃圾短信分类问题的一个有效的方案。



首先读入lintcode网站所提供的训练集数据，按照2.1节所示将所有英文单词标准化，然后按照2.2节内容进行词干提取工作，再将清洗后的数据按2.3节方法转换成向量形式便于建模。在得到了数值向量形式的训练集之后，就可建立logistics回归模型，并对测试集作出预测。

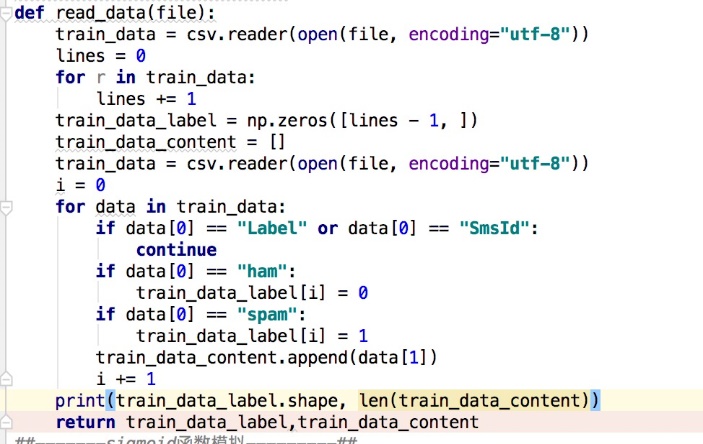
4.2 程序实现

根据4.1节所述的方案，我们利用python实现了这个过程，下面列举了一些关键的程序实现方法。

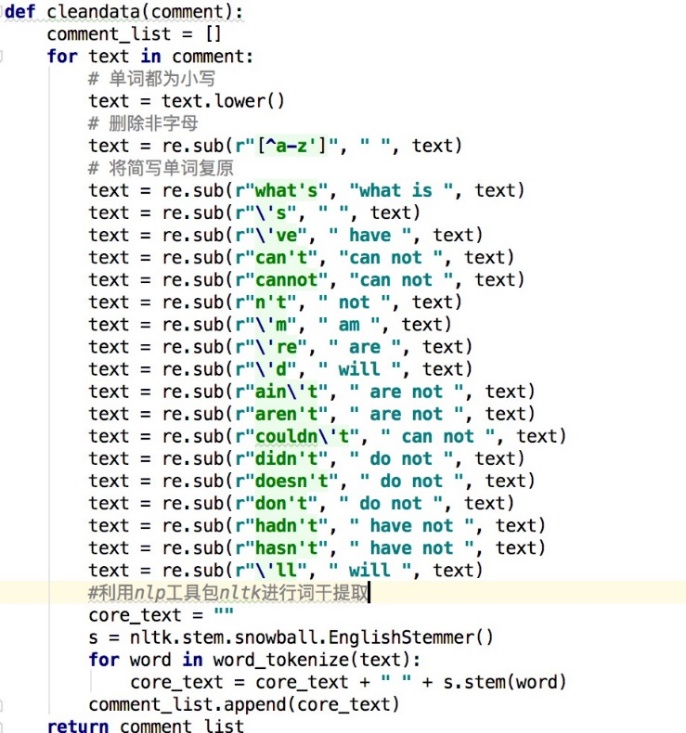


类stocGradAscent1实现了随机选取向量更新权值策略以及步长非严格递减策略，理论上可以使逻辑回归算法迅速收敛。在随机选择向量过程中，从第98行可以看出，利用了random工具随机选择一个向量进行计算并更新权值，最后在第102行从数据集中删掉了这一个向量。值得注意的是，python3必须将dataindex转换成list类型之后才能进行del操作。而在Python2中就不用。第96行说明学习步长存在一个常数项，虽然步长会不断减小，但是最终不会减小到0，同时也保证了学习步长更新时的非严格单调递减性，和第三章中所叙述的方法相同，能够有效地解决了数据波动问题。

类read\_data，作用为读取lintcode垃圾短信分类实验中train\_data.csv中包含的数据，由于其中含有一系列字符串，在用pandas中方法时老是无法读取所有数据，所以这里采用了比较繁琐的按行读取，并且将标签“ham”、“spam”分别表示成了0和1，便于分类。



类cleandata的作用是将训练集中的所有单词变成小写，之后按照常用的英语表达习惯将缩写还原成单独的单词。这是因为在之后的词干提取以及TF-IDF向量转换中，如果为单词为缩写形式则无法对其做判断。



在定义完读取数据的类read\_data和类cleandata之后，可以按照如下的步骤对训练集进行处理。首先调用类read\_data读取训练集、测试集中的数据，之后进行数据清洗，这样train\_data\_content和test\_data\_content中就仅仅保留了利用snowballStemmer方法所保留的词干。之后直接调用TfidfVectorizer函数进行TF-IDF特征转换，转换后的训练集和测试集分别放在train\_x、test\_y中。到这里，将自然语言转化为机器学习算法可以识别的向量的工作就完成了，之后的工作就和普通的对率回归基本一致，即计算出各变量的权值，然后输入sigmoid函数进行分类，以及输出最后的结果。



最终结果显示最好的得分为0.85，排在所有队伍的第52名，得分还不是很好，我想对率回归可以看做一个单层神经网络，如果用BP神经网络进行实验的话，可能效果会更好。



5 总结

本次实验完成了lintcode网站AI题中的垃圾短信分类题目，首先在将所有单词标准化之后，利用snowball方法进行词干的提取，然后利用TF-IDF特征向量转换方法将自然语言转换为数值向量，最后用logistics回归方法进行建模预测。整个方案都是我们小组讨论所得出，大家一起合作完成了程序以及报告。虽然最后参数调了很久，但是成绩还是第一次的参数最好，而且成绩排名也算靠前，但是我们也都对自然语言处理这个陌生的方向有了一定的认识，加深了对模式识别课程中内容的记忆。