#### **文章自动分类打标签**

# 一、项目背景与目标

目标网站上积累了大量与疾病主题相关的文章、帖子等文本数据，这些文章没有做内容分类，也没有打上相应的标签，不便于对内容进行管理。为了对文章按主题进行分类、做集约化管理，方便后期向用户做个性化推荐，需要先将大量历史文章、帖子做分类整理，同时为每篇文章打上与其主题相关的标签。

# 二、思路

首先，将数据源分类为训练集和测试集，分别对其进行分词处理，建立语料库、去除停用词，这里需要使用python中的jieba库做中文分词；

其次，为了方便后续生成词向量空间模型，将分词后的训练集和测试集分别转换成文本向量信息并对象化，需要用到sklearn库中的Bunch数据结构；

再者，分别对训练集和测试集进行特征建立，使用TF-IDF算法计算词空间向量，其中需要将训练集的词向量空间坐标赋值给测试集数据；

 最后，构建模型对文本进行分类与模型评价，需要使用朴素贝叶斯算法。

# 三、算法简介

TF-IDF：该算法是一种针对关键词的统计分析方法，用于评估一个词对于一个文件集或一个语料库的重要程度。它基于假设—一个词的重要程度跟它在文章中出现的次数成正比，与它在语料库出现的频率成反比。这种计算方式能有效避免常用词对关键词的影响，提高关键词与文章之间的相关性。

其中TF=（某词在文档中出现的总次数/文档的词总数），IDF=IMG_256lg(语料库中文档总数/包含该词的文档数)+1；当然该算法也有它的**缺陷**：**第一**，它没有考虑文档中该词本身的**重要性**，**第二**，没有考虑语料库中该词的**分布情况**，例如该词在语料库中某一类文档出现频率很高，在其他类中出现频率低，本质上该词对于文章来说是很重要的，但基于该算法，它的IDF权重系数会很小，认为该词不重要。**第三**，没有考虑到词在文章中的**位置**。

具体可以在sklearn中调用TfidfVectorizer库实现TF-IDF算法，并且可以通过stop\_words参数来设置文档中的停用词（没有意义的词，比如语气词等）使停用词不纳入计算范围。

**朴素贝叶斯分类**

该算法是基于贝叶斯原理，它是一种简单但极为强大的预测建模算法。之所以称为朴素贝叶斯，是因为它假设每个输入变量是独立的，这是一个强硬的假设，实际情况并不一定，但这项技术对于绝大部分的复杂问题仍然非常有效。

朴素贝叶斯模型由两种类型的概率组成：

 1、每个**类别的概率**P()

 2、每个属性的**条件概率**P(|)

为了训练朴素贝叶斯模型，我们需要先给出训练数据，以及这些数据对应的分类。那么类别概率和条件概率，他们都可以从给出的训练数据中计算出来，一旦计算出来，概率模型就可以使用贝叶斯原理对新数据进行预测。本案例使用sklearn中的MultinomialNB包实现对文章的朴素贝叶斯分类。

# 四、建模流程

**1、文本分词处理**

为了构建词空间向量，首先需要对待分类文本做切词处理，将切好后的词语写入指定的路径下。这里我们使用python中的jieba工具对文本进行分词，同时使用jieba.analyse.extract\_tags方法（基于TF-IDF算法）抽取文章的主题标签。代码文件（word\_cut.py）

执行程序后，将训练集和测试集对应的文本分别进行了分词处理，并将分词后的文本存入了新建立的文件夹中（train\_segments,test\_segments），同时对测试集的每一个文本抽取了文章的主题标签，存入了新文件夹（theme\_tag1）。

**2、数据结构处理**

为了方便后续生成词向量空间模型，需要将这些分词后的文本信息转换成文本向量信息并对象化。这里使用sklearn库中Bunch数据结构，将文本存储成链式结构，本文中定义Bunch(label=[],filepath=[],contents=[])，其中label标注训练集每份文本归属的类别，filepath标注文本的存储路径，contents保存训练集和测试集中每一类别下的文本内容。代码（文件word\_to\_bunch.py）。

上述程序是将每篇文档的类型、存储路径、文章内容写入Bunch数据结构中，执行程序后生成train\_bunch\_bag.dat和test\_bunch\_bag.dat文件。

**3、计算文本的TF-IDF权重矩阵**

此步骤是将上一步存储的结构化数据构建成为一个TF-IDF词向量空间，空间中的词全部来自该训练集，各个词的权重矩阵也一并保存下来，建模过程中需要注意将训练集的词向量空间的坐标赋值给测试集。代码（文件TF-IDF\_count.py）。

执行该程序会将训练集和测试集数据转换为TF-IDF词向量空间中的实例，其中space表示词向量空间坐标，tdm表示训练集合测试集数据的TF-IDF权重矩阵。执行完程序生成train\_tfidfspace.dat和test\_tfidfspace.dat数据文件。

**4、朴素贝叶斯方法分类文章**

使用训练集数据训练模型得到朴素贝叶斯分类器，再用训练好的分类器对测试集进行分类，得到结果，从精确率、召回率、F-测度值三个角度评价。代码（nbayes\_classify.py）。

从得到的结果可以看出，精确率0.926，召回率0.923，F-测度值0.933，分类效果还是很不错的，同时将预测结果写入新文件（classify\_result\_file.txt）。