# 说明文档

[说明文档 1](#_Toc21847)

[1. 项目概述及人员： 2](#_Toc20446)

[1.1 项目概述 2](#_Toc11590)

[1.2 人员 2](#_Toc25152)

[2. 需求分析 2](#_Toc32556)

[3. 系统设计 3](#_Toc19084)

[3.1 总体设计 3](#_Toc16628)

[3.2 系统功能模块 3](#_Toc22357)

[4. 测试及分析 4](#_Toc24017)

[4.1交叉验证的方法 4](#_Toc2607)

[4.2交叉验证结果 4](#_Toc12176)

[4.3结果分析 4](#_Toc16941)

[4.4可优化地方 4](#_Toc15241)

[参考文献 4](#_Toc20369)

[结束语 4](#_Toc6162)

## 项目概述及人员：

### 1.1 项目概述

实现一个分类系统。

分类体系为：财经(类别号：1)、科技(类别号：2)、汽车(类别号：3)、房产(类别号：4)、体育(类别号：5)、娱乐(类别号：6)、其它类(类别号：7)，利用网站的新闻主页(可以下载Sogou语料)，训练一个分类器(训练集合不能少于5000篇文档)。

结果：实现新的网页的分类。支持交互式URL输入，或者输入一个文本，文本每行都是一个URL，系统输出结果文本，每行对应输入文本的类别号。

### 1.2 人员

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 分工 |
| 冯云 | 201618018629135 | 爬虫编写 |
| 苏俊玮 | 2016E8018661180 | 界面编写 |
| 苏莉娅 | 201628018629065 | 分类器编写文档整理 |
| 尹捷 | 201628018627124 | 爬虫编写ppt整理 |
| 张晓欧 | 2016E8018661198 | 界面设计文档整理 |

## 需求分析

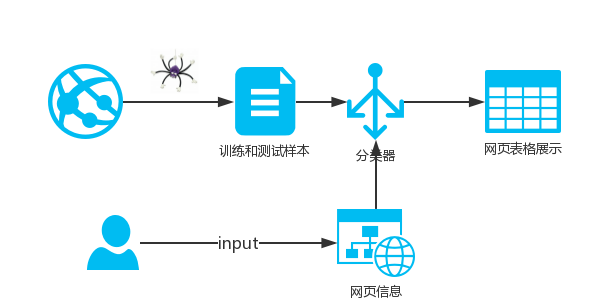
1. 功能需求：新闻或资讯展示平台需要将得到的最新资讯进行自动分类处理，以快速的发布更新资讯。

支持交互式提交新的网站链接，得到分类器分析结果。

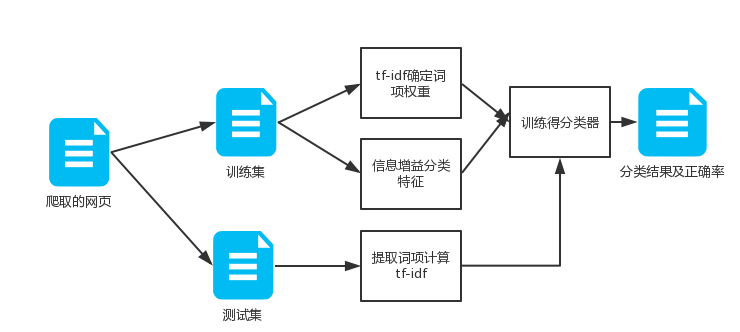
（2）性能需求：爬取数据要保证速度的同时不被封号；分类器分类结果的正确率要在60%以上为优。

## 3. 系统设计

### 3.1 总体设计



图一：系统总设计图



图二：分类器设计图

系统分为生成分类器和结果测试两个部分。

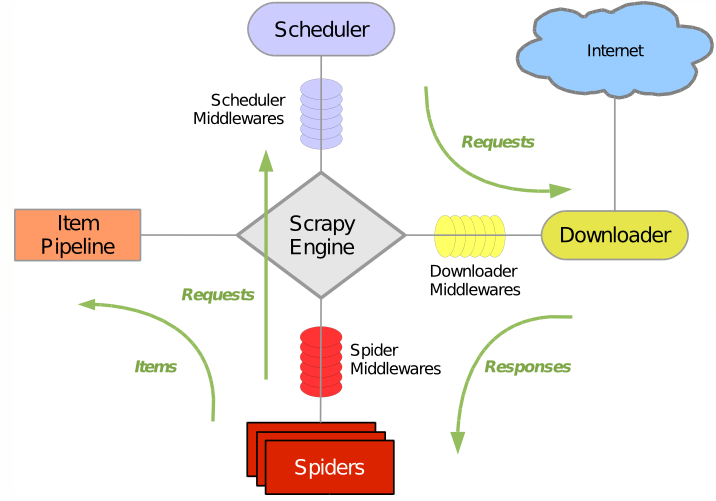
生成分类器包括：训练和测试两个部分。训练部分采用的主要方法为KNN和贝叶斯两个方法，使用信息增益为特征，根据tf-idf设置权重。

结果测试部分通过网页与用户交互，用户在网页上提交网页链接，系统自动调用分类器，通过条形图可视化的展示分类的效果。

### 3.2 系统功能模块

#### 3.2.1 爬取部分

1. 爬虫原理：



图三：Scrapy整体架构

首先获取第一个URL的初始请求，当请求返回后调取一个回调函数。第一个请求是通过调用start\_requests()方法。该方法默认从start\_urls中的Url中生成请求，并执行解析来调用回调函数。

在回调函数中，你可以解析网页响应并返回项目对象和请求对象或两者的迭代。这些请求也将包含一个回调，然后被Scrapy下载，然后有指定的回调处理。

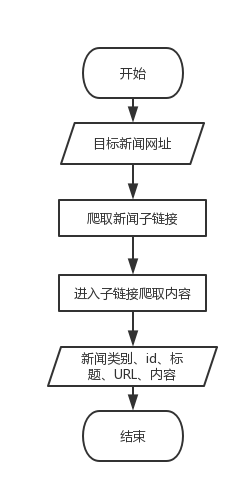
在回调函数中，你解析网站的内容，同程使用的是Xpath选择器（但是你也可以使用BeautifuSoup, lxml或其他任何你喜欢的程序），并生成解析的数据项。

最后，从蜘蛛返回的项目通常会进驻到项目管道。

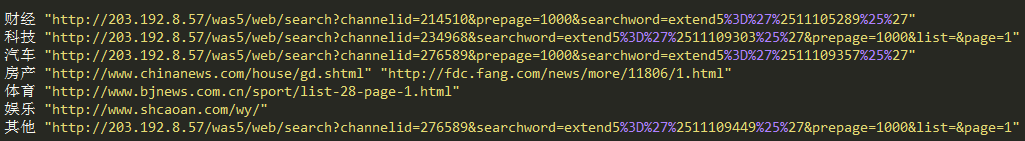
2、具体实现：

在新闻爬取部分，由于不同类别的新闻来源有所不同，网页数据提取方式不同，因此对每种类别的新闻分别进行爬取。

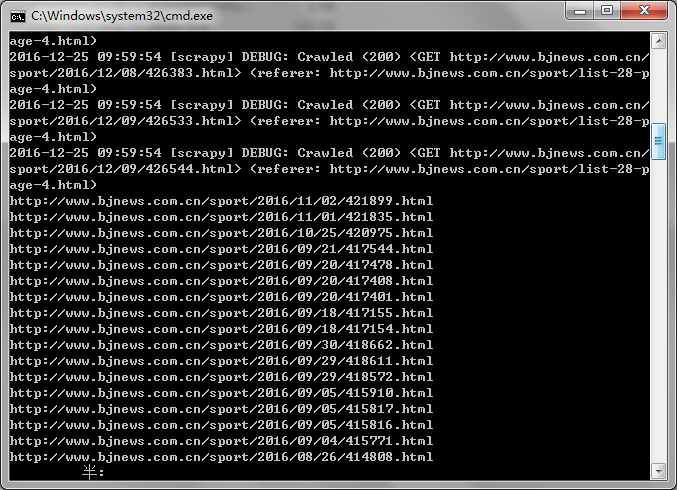
利用Scrapy框架，首先在程序中定义目标新闻列表页面，用XPath进行网页数据提取，从初始页面爬取单条新闻子链接，然后对每个新闻子链接进行爬取，获得新闻标题及内容，按照{类别号，id号，标题，URL，内容}的格式储存为JSON文件。



图三：爬取的流程如下图



图四：新闻爬取地址



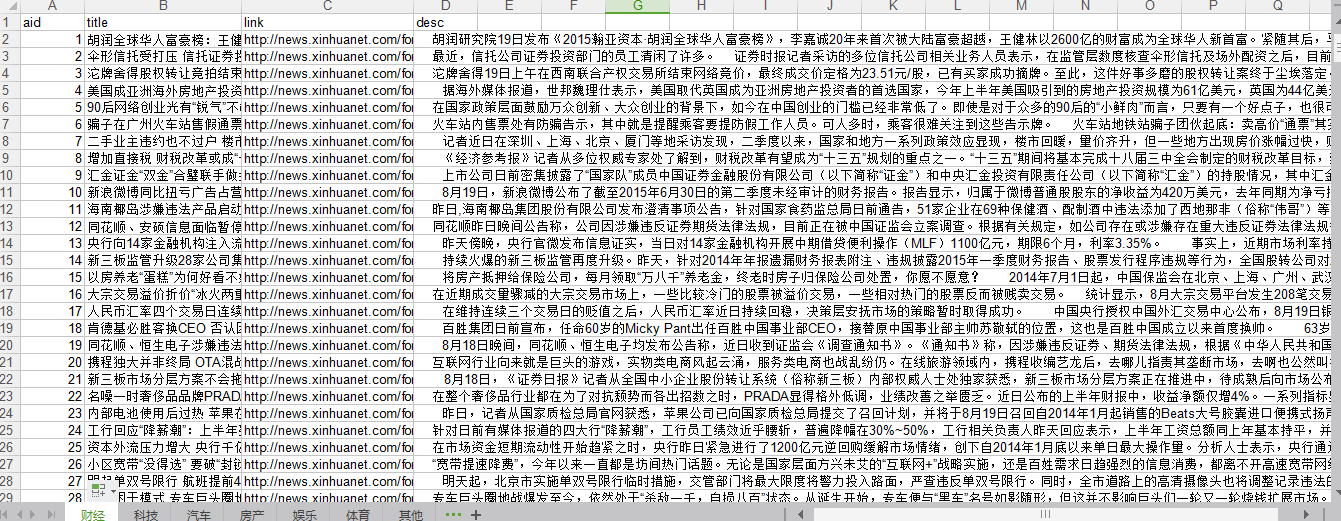
图五：运行示意图



图六：得到的JSON文件

共爬取财经类新闻992条，科技类新闻999条，汽车类新闻330条，房产类新闻584条，体育类新闻1000条，娱乐类新闻485条，其他类新闻1000条。

然后，为方便后续分类器的训练，编写j2e.py脚本将JSON文件内容储存到Excel表格中，每个类别的新闻分别放在一个标签页中，得到news.xlsx，如下图所示：



图七：excel表格存储结果

#### 3.2.2 分类器部分

包括输入输出的处理流程图。分类原理。实现过程遇到的问题。可优化地方。运行截图和说明。

1、分类原理：

kNN的分类原理：

k近邻法(k-nearest neighbor. k-NIs)是一种基本分类与回归方法，k近邻法的输入为实例的特征向量，对应于特征空间的点；输出为实例的类别，可以取多类。k近邻法假设给定一个训练数据集，其中的实例类别己定。分类时，对新的实例，根据其k个最近邻的训练实例的类别通过多数表决等方式进行预测。因此，k近邻法不具有显式的学习过程。k近邻法实际上利用训练数据集对特征向量空间进行划分，并作为其分类的“模型”。k值的选择、距离度量及分类决策规则是k近邻法的三个基本要素。

k值的选择会对k近邻法的结果产生重大影响。k值的减小就意味着整体模型变得复杂，容易发生过拟合；如果选择较大的k值，就相当于用较大邻域中的训练实例进行预侧，其优点是可以减少学习的估计误差。但缺点是学习的近似误差会增大。在应用中，**k值一般取一个比较小的数值。通常采用交叉验证法来选取最优的k值。**

特征空间中两个实例点的距离是两个实例点相似程度的反映。k近邻模型的特征空间一般是n维实数向量空间Rn。本次我们的程序中使用的是**欧氏距离**。

k近邻法中的分类决策规则往往是**多数表决**，即由输入实例的k个邻近的训练实例中的多数类决定输入实例的类。

贝叶斯分类器原理：

朴素贝叶斯(native Bayes)法是基于贝叶斯定理与特征条件独立假设的分类方法。对于给定的训练数据集，首先基于特征条件独立假设学习输入/输出的联合概率分布；然后基于此模型，对给定的输入x，利用贝叶斯定理求出后验概率最大的输出y。朴素贝叶斯法 实现简单，学习与预测的效率都很高，是一种常用的方法。

利用训练数据学习P(X |Y)均和P(Y)的估计，得到联合概率分布：

P(X，Y) = P(Y)P(X|Y)

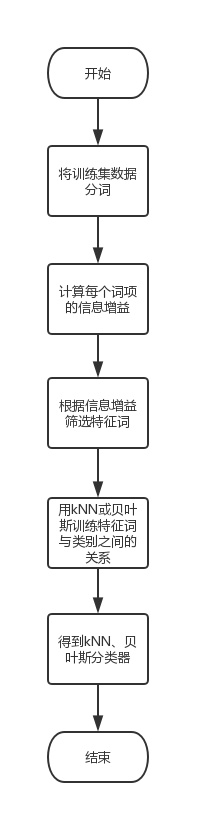
概率估计方法可以是极大似然估计或贝叶斯估计。

朴素贝叶斯法的基本假设是条件独立性，这是一个较强的假设。由于这一假设，模型包含的条件概率的数量大为减少，朴素贝叶斯法的学习与预测大为简化。因而朴素贝叶斯法高效，且易于实现。其缺点是分类的性能不一定很高。

2、具体实现：

网页分类主要分为两部分实现：分类器构建和网页分类。在分类器构建中，首先将爬取到的数据进行分词，然后计算词的信息增益选择特征词，将这些特征词用两种分类方式进行训练（kNN和贝叶斯），得到两种分类方法的分类器；在网页分类中，将网页的文本首先进行分词，计算tf-idf值，选择文本的“关键词”，使用这些词在分类器中的信息增益和词在待查文档中的权重来测试判定网页类别。

1）分类器构建



分类器的构建流程图

2）文本jieba分词

首先将得到的新闻文本字符串进行分词，使用jieba分词工具。本次用到了jieba分词中适合文本分析的精确模式。在分词的过程中，我们删除了单字词语和标点符号。因为单字词语通常是无意义的“的”、“地”等词语，标点符号并不能做为文本分类的根据。通过分词，我们得到分好词的文本列表wordlist。

3）计算词项信息增益

计算上步得出的分词的信息增益，首先计算局部类别文档频率mdf（分词后的词项在该类别中的文档频率）和全局文档频率ndf（分词后的词项在全局文档中的概率），词项i的信息增益计算如下：

其中，参数分别表示：：词项在所有文档中出现的概率

：词项在所有文档中未出现的概率

：词项在该类别文档中出现的频率

：词项在其他类别文档中出现的频率

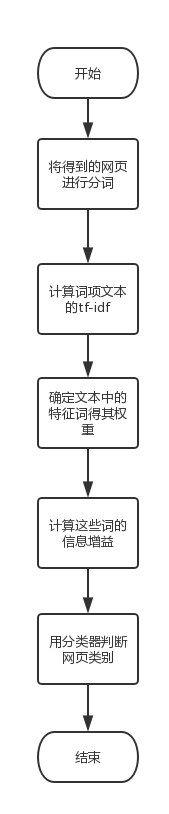
：词项未在该类别文档中出现的频率

：词项未在其他类别文档中出现的频率

筛选特征词

将计算得到的词项的信息增益进行比较，选择信息增益较高的词作为特征词。

4）网页分类



网页分类的流程图

1. 将网页内容进行分词
2. 确定文本的特征词得其权重
3. 计算tf-idf

首先将得到分词之后的文本字符串的向量化表示，进行归一化处理，得到词项频率tf的字典形式（一篇文档对应一个字典）；

然后根据类别文档频率mdf和全局文档频率ndf计算每一类文档中各个词条的改进的idf权重：

得到每个元素为该类别中的词条的权重列表，按权重由大到小排列，元素为元组，元组的第一个元素单词，第二个元素表示该单词在该类别中的权重；

最后根据上两步得到的tf、idf计算每篇文档的tf-idf权重向量。

1. 计算特征词的信息增益

从上一步可以看出词的权重，从而提取特征词，计算这些特征词在文本中的信息增益。

1. 用分类器判断网页类别

将上一步的特征词的信息增益带入kNN或贝叶斯的分类器中，得到网页类别。

3、实现中遇到的问题

1）分类器的分类方法选择

开始试图尝试了决策树和神经网络反馈方法进行分类，但是觉得效率不高而且分类效果并不好，最后选择了kNN和贝叶斯两种分类方法。

2）特征词提取的数目

特征词提取的数目是直接影响正确率的，如果特征词提取过少，可能分类的特征就不会很清晰，如果特征词提取过多，可能会出现分类结果与测试集过分拟合的问题，不符合实际情况。最终我们选取25个特征词来代表待分类文本。

4、优化方法

1）使用多中分类方法交叉对比

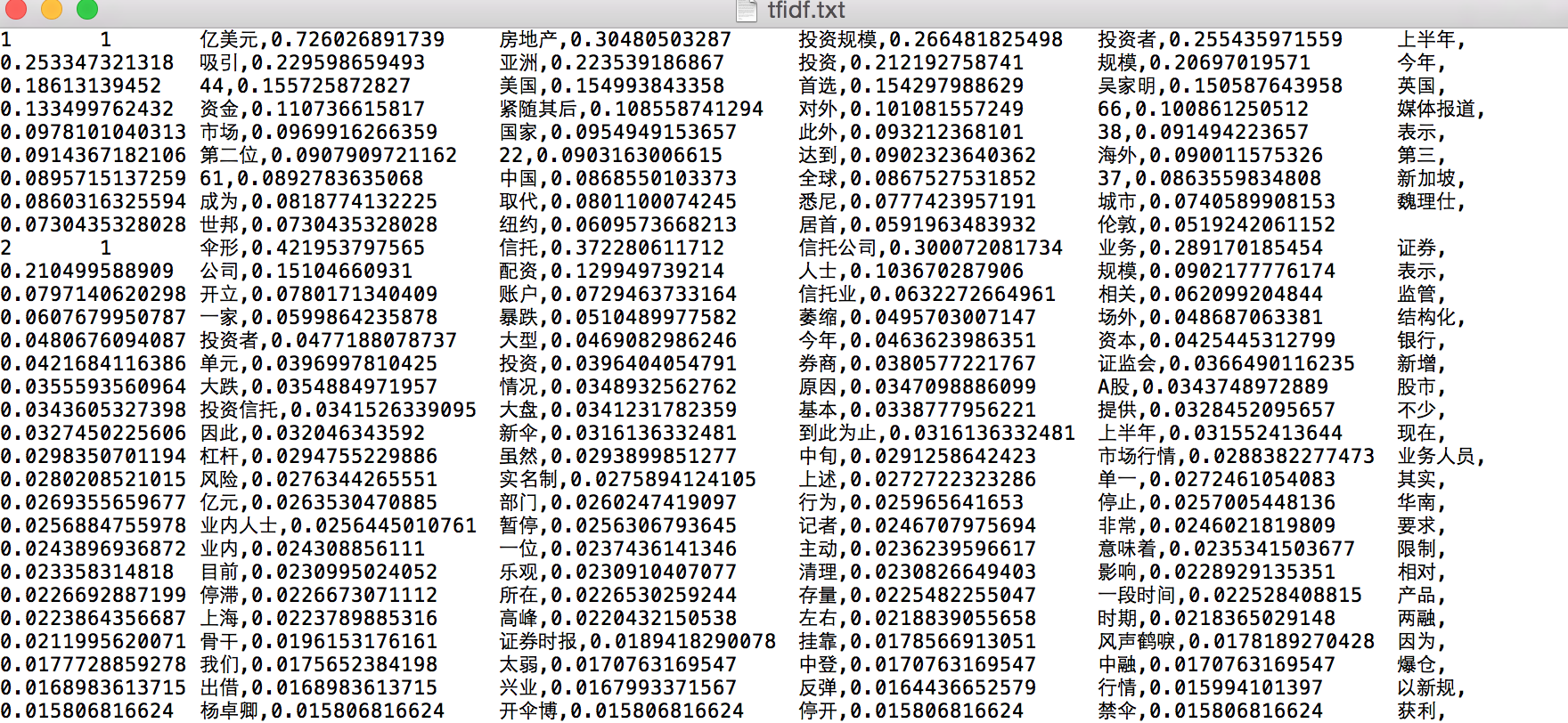
可以使用多种分类方法交叉对比构建分类器，将分类器构建的分类结果进行比对，分类结果的正确性会提高。

2）实现k近邻法时，主要考虑的问题是如何对训练数据进行快速k近邻搜索。这点在特征空间的维数大及训练数据容量大时尤其必要。当训练集很大时，计算非常耗时，这种方法是不可行的。为了提高k近邻搜索的效率，可以考虑使用特殊的结构存储训练数据，以减少计算距离的次数。

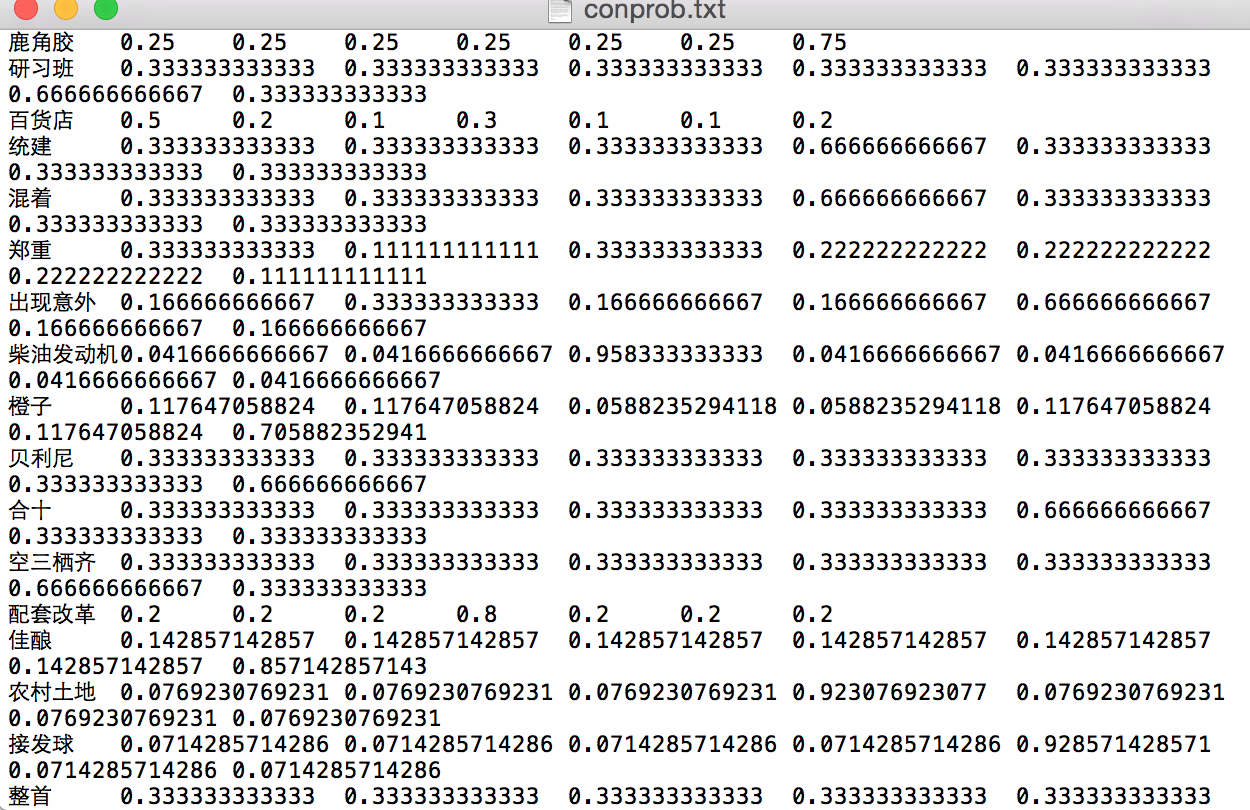
#### 3.2.3 功能集成

1）通过数据爬取模块获取训练集，包括新闻的链接、标题和内容，详见爬取模块。

2）对爬取的数据进行分析。训练集使用结巴分词（jieba）插件对文本分词，去除符号和长度为1的词。（featureextract.py. tokentext）；然后计算每个文档中每个单词的词项出现的次数tf; 计算文档集合里的df、idf，这里使用改进算法，idf=log{N\*(m/n)\*(m/(n-m+1))}；计算td-idf，并存储成为td-idf.txt；计算信息增益ig和条件概率，并存储为conprob.txt



td-idf.txt

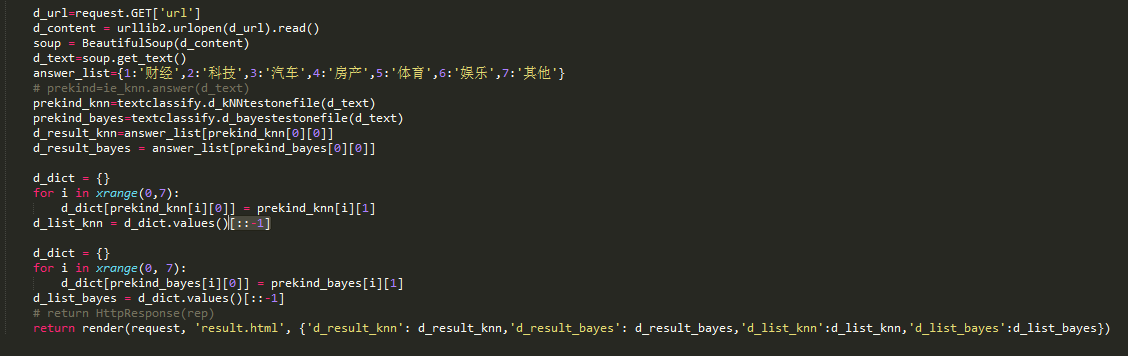


conprob.txt

3）我们选用djongo框架来处理web数据并提供分类视图显示。访问127.0.0.1:8000 显示index页面。输入待测URL，使用urllib2.urlopen(链接).read()读取网页的内容。并使用bs4插件中的BeautifulSoup类读取网页中的文本内容。

4）调用分类器模块中kNNtestonefile 和bayestestonefile 函数对文本内容进行分类处理，并将选取最终排名结果赋予变量d\_result\_knn和d\_result\_bayes

5）将分类后得到的各类别的得分即prekind\_knn和prekind\_bayes二维数组中的值取出来存成字典，并排序。最终，将给类别的隶属度得分以及分类结果显示到result.html页面中。



#### 3.2.4 展示模块：

1、输入输出流程：

1）输入待测的URL

2）爬取网站中的主体文本。即接收用户输入的链接，使用urllib2.urlopen(链接).read()读取网页的内容。并使用bs4插件中的BeautifulSoup类读取网页中的文本内容。

3）对网站文本进行分词处理。

4）分别使用贝叶斯和KNN算法对文本分词数据进行分析，分别计算其权重和余弦相似度。最终得到排名最高的类别。

5）输出分类结果



1. 使用过程展示

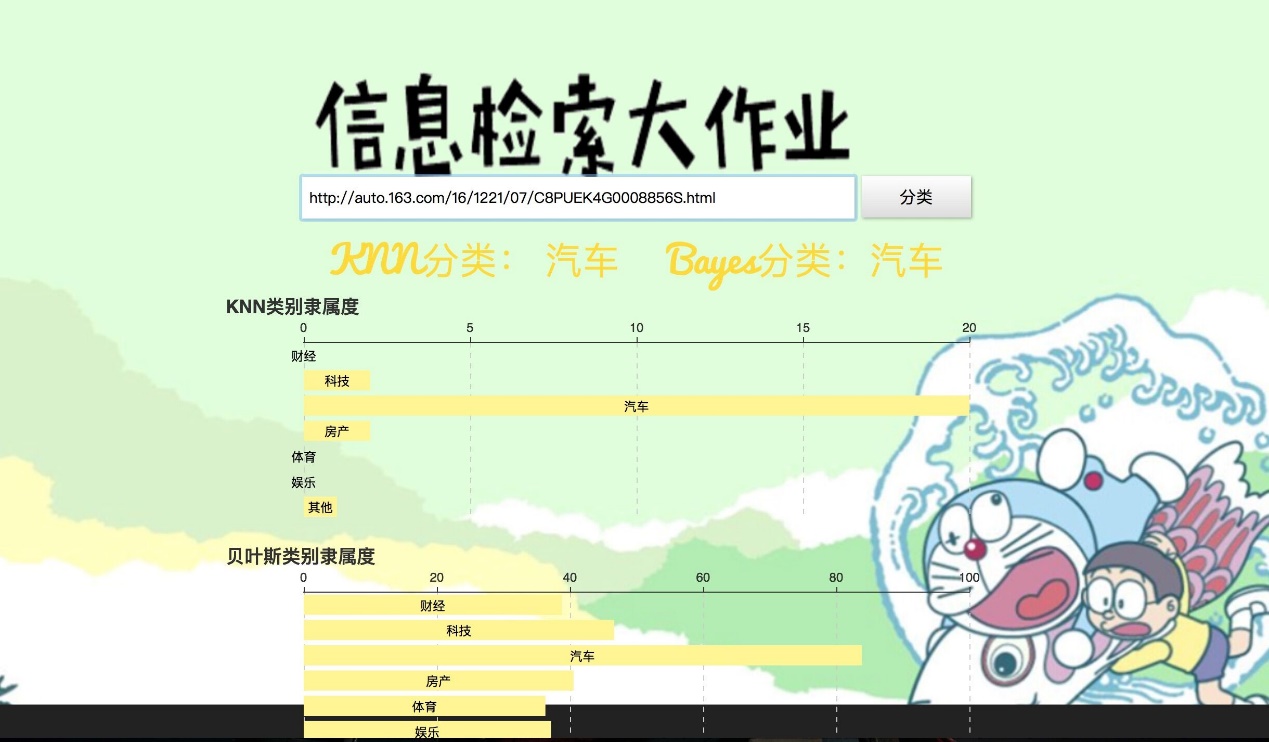
首页展示：



找到一篇与汽车相关的文档：



将其URL输入搜索框，点击分类，得到结果。



结果显示：利用KNN和贝叶斯算法均能将该文档准确分类为汽车类别。

## 4. 测试及分析

### 4.1交叉验证的方法

交叉验证（Cross Validation）是用来验证分类器的性能一种统计分析方法，基本思想是把在某种意义下将原始数据（dataset）进行分组，一部分做为训练集（training set），另一部分做为验证集（validation set），首先用训练集对分类器进行训练，在利用验证集来测试训练得到的模型（model），以此来做为评价分类器的性能指标。

**其中K-fold Cross Validation（K-折交叉验证，记为K-CV）：**将原始数据分成K组（一般是均分），将每个子集数据分别做一次验证集，其余的K-1组子集数据作为训练集，这样会得到K个模型，用这K个模型最终的验证集的分类准确率的平均数作为此K-CV下分类器的性能指标。K一般大于等于2，实际操作时一般从3开始取，只有在原始数据集合数据量小的时候才会尝试取2。K-CV可以有效的避免过学习以及欠学习状态的发生，最后得到的结果也比较具有说服性。

### 4.2交叉验证结果

在实际的训练中使用了7次交叉验证，分别测试7次实验的正确率，最终得到平均效果的正确率，如图所示：



### 4.3结果分析

从结果可以看到，对于财经类的分类效果非常好，房产类体育类娱乐类由于训练集相关数据的缺乏导致正确率不高。

### 4.4可优化地方

1、爬虫爬取数据的类型需要进一步丰富， 以支持多类别的数据分类。爬虫在爬取大量数据的时候容易出现被封现象，针对这一点可以进一步从爬虫频率，使用代理等角度进行优化。

2、如果处理更多的数据，需要对贝叶斯和KNN实现算法实现过程进行优化，可考虑使用特殊的结构存储训练数据，以减少计算的次数。

## 5结束语

本项目通过对网页数据的多种分类方法尝试，发现贝叶斯和KNN分类方法对于此类数据有较好的分类效果，符合实验需求的要求。通过算法实现的过程，对于信息检索的实现过程有了深刻的理解，特别是针对本实验中的数据分类结果针对不同类别分类效果差别很大做了深入的探讨和分析。对于优化算法、提高所有类别分类效果做了很多尝试，深入体会了困难所在，以及一些初步的优化想法，例如通过改善爬虫爬取数据的方法增加训练集的丰富性，通过采用更多的数据特征的方式提高分类效果等。

项目已在github上公开，后续将结合相关工作做进一步完善。