

# 第八届“泰迪杯” 全国数据挖掘挑战赛

『智慧政务』中的文本挖掘与分析应用

# “智慧政务”中的文本挖掘与分析应用

## 摘要

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升。同时，随着大数据、云计算、人工智能等技术的发展，运用网络文本分析和数据挖掘技术对相关文本的研究具有重大意义。

针对问题 1，首先对附件 2 的群众留言数据进行数据预处理。利用 jieba 中文分词工具对留言主题信息进行分词，并去除停用词，通过 TF-IDF 算法来提取留言主题信息的特征值，生成每个留言主题短文本的特征向量。本题选择支持向量机分类器去训练模型，建立关于留言主题的一级标签分类模型，通过模型去预测新数据的类别，最后通过 F-Score 对分类方法进行评价。

针对问题 2，通过利用获取 Excel 中“留言详情”的一列，对要计算的多行内容进行分词、去除停用词、获取词频、词袋模型向量化文本、LSI 模型向量化文本，最后再计算相似度。经过多次训练模型主题，再结合附件 1 中一级分类的内容，可得出最终模型主要可分为 15 个热点。根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，统计出同类热点的留言总数定义为热度评价指标，根据词频的概率给出相应的评价结果。

针对问题 3，首先读取附件 4 的数据，并将数据写入到 txt 文本中，方便我们后续的操作。然后我们再读取存入到 txt 文本，在网上下载一篇较为科学合理的停用词文本，利用 jieba 中文分词工具对文本进行分词和去除停用词。接着还是使用 jieba 提取留言详情和答复意见的关键词，将提取到关键词写入到 txt 本文中。最后用 gensim 对留言详情和答复意见的关键词做相似度分析。

**关键词：**支持向量机；词袋模型；LSI 模型；向量化文本；TF-IDF 算法

# Text mining and analysis in "smart government"

## **abstract**

In recent years, with wechat, Weibo, mayor's mailbox, sunshine hotline and other online political platforms gradually becoming an important channel for the government to understand the public opinion, gather people's wisdom and gather people's spirit, the amount of text data related to various social situations and public opinions is increasing. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the use of network text analysis and data mining technology is of great significance to the research of related texts.

To solve the problem 1, we first preprocess the data of the mass message in attachment 2. Using the Chinese word segmentation tool of Jieba to segment the message subject information and remove the stop words, TF-IDF algorithm is used to extract the feature value of message subject information and generate the feature vector of each message subject short text. In this paper, support vector machine classifier is selected to train the model, and a first-class label classification model about message topic is established, through which new data categories are predicted. Finally, the classification method is evaluated by F-score.

To solve the problem 2, we use the column of "message details" in Excel to segment the multi line content to be calculated, remove the stop words, obtain word frequency, word bag model vectorized text, LSI model vectorized text, and finally calculate the similarity. After training the subject of the model several times, combined with the content of the first level classification in Annex 1, the final model can be divided into 15 hot spots. According to Annex 3, we classify the messages that reflect the problems of specific places or specific groups of people in a certain period of time, and count the total number of messages of the same hot spots as the heat evaluation index, and give the corresponding evaluation results according to the probability of word frequency.

To solve the problem 3, first read the data of attachment 4 and write the data into

the TXT text, which is convenient for our subsequent operation. Then we read and save the text into TXT, download a more scientific and reasonable stop words text on the Internet, and use the Chinese word segmentation tool of Jieba to segment and remove the stop words. Then we use Jieba to extract the key words of message details and reply comments, and write the extracted key words into the TXT text. Finally, we use gensim to analyze the similarity between the details of the message and the key words of the reply.

**Keywords:** support vector machine; word bag model; LSI model; vectorized text; TF-IDF algorithm

# 目录

1.挖掘目标.....	1
2.分析方法与过程.....	2
2.1 问题 1 分析方法与过程.....	2
2.1.1 流程图.....	2
2.1.2 数据预处理.....	4
2.1.3TF-IDF 文本特征选择 .....	6
2.1.4 支持向量机（LinearSVC） .....	7
2.1.5 评价指标.....	8
2.2 问题 2 分析方法与过程.....	9
2.2.1 数据的选取.....	9
2.2.2 自定义词典与词性.....	10
2.2.3 分词及去除停用词.....	10
2.2.4 词频计算.....	11
2.2.5 制作语料库.....	12
2.2.6 建立 LSI 模型.....	13
2.2.7 相似度分析.....	13
2.3 问题 3 分析方法与过程.....	14
2.3.1 问题 3 流程图.....	14
2.3.2 文本预处理.....	14
2.3.3 提取关键词 <sup>[8]</sup> .....	16
2.3.4TF-IDF 算法 <sup>[9]</sup> .....	16
2.3.5 关键字相关性.....	17
3.结果分析.....	18
3.1 问题 1 结果分析.....	18
3.1.1 分类结果评估.....	18
3.2 问题 2 结果分析.....	19
3.2.1 对热点问题表分析.....	19
3.2.2 对热点问题留言明细表分析.....	22
3.3 问题 3 结果分析.....	23
3.3.1 相关性分析.....	23
3.3.2 完整性分析.....	24
3.3.3 可解释性分析.....	24
4.结论 .....	26
5.参考文献.....	27

## 1.挖掘目标

本次建模通过微信、微博、市长信箱、阳光热线等网络问政平台获取到的各类社情民意相关的文本数据信息，利用 jieba 中文分词工具和提取关键词工具，TF-IDF 算法，达到以下三个目标：

（1）利用文本分词、TF-IDF 以及支持向量机的方法对非结构化的数据进行文本挖掘，建立关于留言内容的一级标签分类模型，以便后续将群众留言分派至相应的职能部门处理，提升政府的管理水平和施政效率。

（2）及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，输出排名前 5 的热点问题，保存为文件“热点问题表.xls”，和输出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

（3）根据附件 4 相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出评价。

## 2.分析方法与过程

### 2.1 问题 1 分析方法与过程

#### 2.1.1 流程图

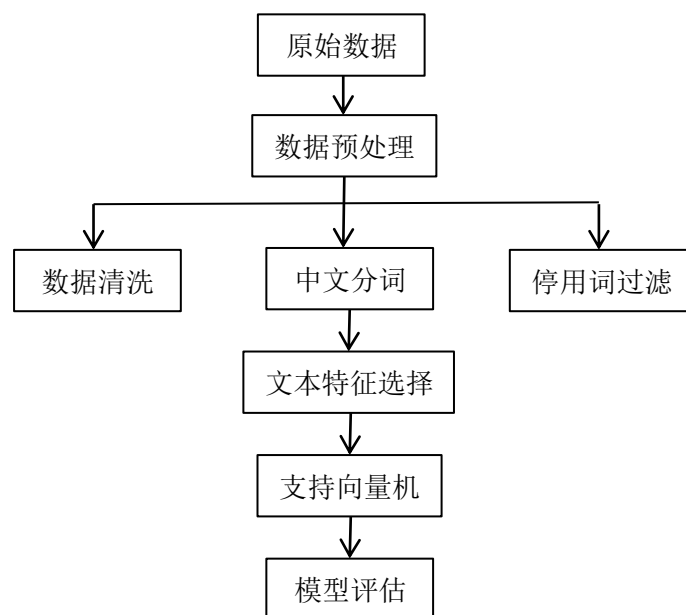


图 1 问题 1 流程图

机器学习项目有（分类/回归/聚类）三大类问题，分类就是训练集已知样本和对应的类别，然后建立模型去预测新的样本类别并评估模型的优劣；回归就是相当于分类中的离散值类别变成连续值；而聚类是训练集中没有类别，需要通过聚类算法去找到对应类别。

附件 2 数据中包含了 7 个一级分类标签（城乡建设、环境保护、交通运输、教育文体、劳动和社会保障、商贸旅游和卫生计生），共 9210 条群众留言数据，每个类别下的文本数量如表 1 所示，类别分布直观图如图 2 所示。我们想要把不同群众留言数据分到不同的分类中去，且每条数据只能对应 7 个类中的一个类。

表 1 数据集各类别留言

类别	数量
城乡建设	2009

劳动和社会保障	1969
教育文体	1589
商贸旅游	1215
环境保护	938
卫生计生	877
交通运输	613

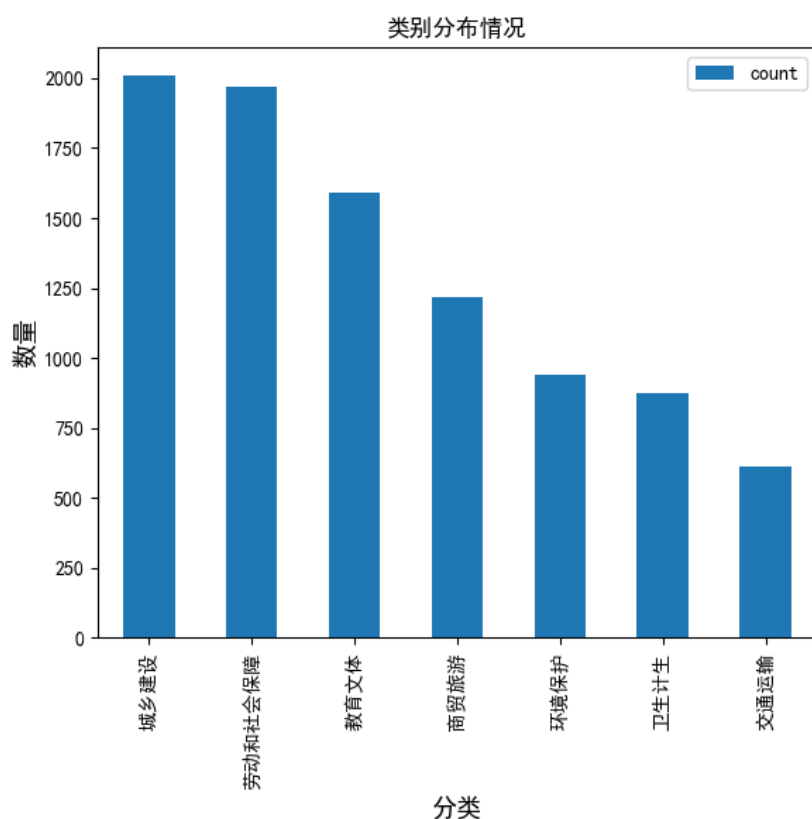


图 2 类别分布直观图

由此可见问题 1 是一个典型的分类问题，由于留言用户所提的问题主题和内容都很短，所以这是一个短文本分类问题，确定好问题之后，问题 1 的分析流程可大致分为以下几步：

第一步：下载需要分析使用的原始数据（附件 2）。

第二步：对原始数据进行基本的处理，包括数据清洗、中文分词、停用词过滤等操作。

第三步：留言文本数据经过处理后，运用 TF-IDF 算法进行文本特征选择。

第四步：将文本特征选择之后合成的矩阵作为支持向量机的输入，通过训练



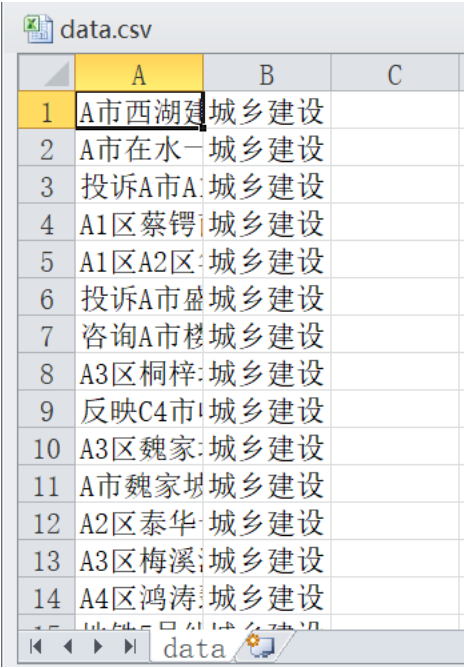
集训练模型，测试集评估分类效果。

2.1.2 数据预处理

(1) 数据清洗

在实际的中文文本分类问题中，我们面对的原始中文文本数据经常会存在许多影响最终分类效果的部分，这部分数据或文本都需要在文本分类最开始的时候就被清洗干净，否则很容易导致所谓的“Trash in, trash out”问题。除了一般分类问题的数据清洗都包含的缺失值处理、去重处理和噪声处理等步骤之外，本题的中文短文本分类还应该处理长串数字或字母。

在题目给出的数据中，还出现了很多建模过程中不需要的列数据。例如留言编号、留言用户、留言时间。因此在对数据分词之前，我们先把不相关的列数据删除。处理后的数据保存在 data.csv 中，如图 3 所示：



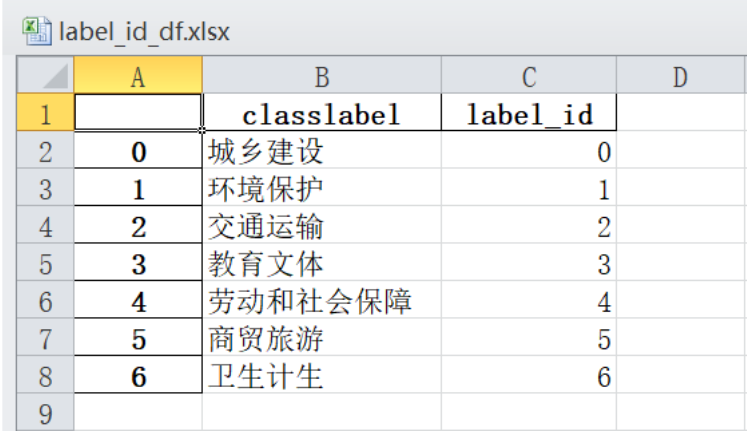
	A	B	C
1	A市西湖区	城乡建设	
2	A市在水一	城乡建设	
3	投诉A市A	城乡建设	
4	A1区蔡锷	城乡建设	
5	A1区A2区	城乡建设	
6	投诉A市盛	城乡建设	
7	咨询A市楼	城乡建设	
8	A3区桐梓	城乡建设	
9	反映C4市	城乡建设	
10	A3区魏家	城乡建设	
11	A市魏家坝	城乡建设	
12	A2区泰华	城乡建设	
13	A3区梅溪	城乡建设	
14	A4区鸿涛	城乡建设	
15	地铁5号线	城乡建设	

图 3 数据清洗后的文本

(2) 中文分词

在对群众留言信息进行数据挖掘分析之前，需要先把非结构化的文本信息转换成计算机能够识别的机构化信息。在附件 2 中，以中文文本的方式给出了数据。为了便于转换，先将中文类别转换成数字类别，如图 4 所示。再对留言内容进行中文分词，这里采用 python 的中文分词包 jieba 进行分词。分词是将连续的字序

列按照一定的规范重新组合成词序列的过程。例如“交通问题”这个词就不能拆成“交通”和“问题”两个词，因此对语料库的更新和补充极其重要，分析时若出现了语料库中没有的词，我们必须将其记录并添加入语料库中，使得能更好的实现中文分词效果<sup>[1]</sup>。



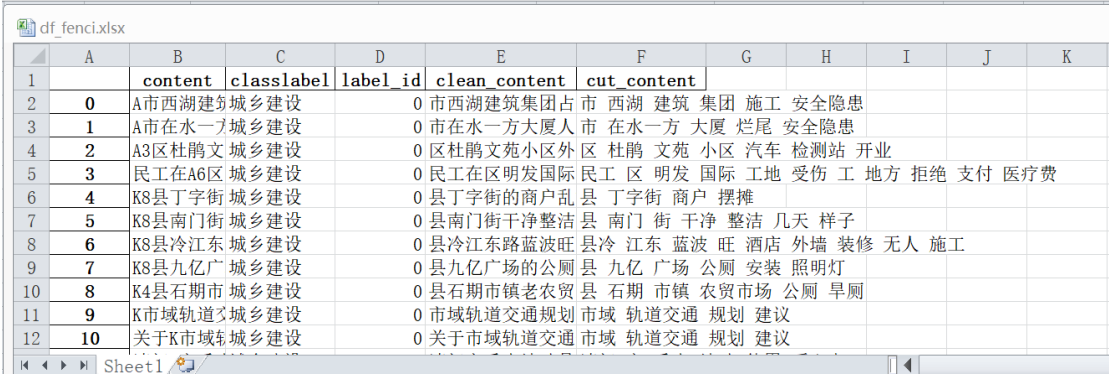
	A	B	C	D
1		classlabel	label_id	
2	0	城乡建设		0
3	1	环境保护		1
4	2	交通运输		2
5	3	教育文体		3
6	4	劳动和社会保障		4
7	5	商贸旅游		5
8	6	卫生计生		6
9				

图 4 文本类别对应的数字类别

(3) 停用词的过滤

停用词是指不包含或包含极少语义，不能反映主题的功能词。例如：“的”、“地”、“得”之类的助词，以及像“然而”、“因此”等只能反映句子语法结构的词语，它们不但不能反映文献的主题，而且还会对关键词的抽取造成干扰，有必要将其滤除。另外标点符号也可以被认为是一种停用词。在文本中去掉这些停用词能够使模型更好地去拟合实际的语义特征，从而增加模型的泛化能力。

本题中，分词以及过滤停用词后的数据保存在 df\_fenci.csv 中，可以看到标点符号、感叹词、数字以及字母等类型的词已经过滤。经过分词以后我们生成了 cut\_review 字段。在 cut\_review 中每个词语中间都是由空格隔开，部分结果如图 5 所示：



	A	B	C	D	E	F	G	H	I	J	K
1		content	classlabel	label_id	clean_content	cut_content					
2	0	A市西湖建	城乡建设	0	市西湖建筑集团占	市 西湖 建筑 集团 施工 安全隐患					
3	1	A市在水一方	城乡建设	0	市在水一方大厦市	在水一方 大厦 烂尾 安全隐患					
4	2	A3区杜鹃文	城乡建设	0	区杜鹃文苑小区外	区 杜鹃 文苑 小区 汽车 检测站 开业					
5	3	民工在A6区	城乡建设	0	民工在区明发国际	民工 区 明发 国际 工地 受伤 工 地方 拒绝 支付 医疗费					
6	4	K8县丁字街	城乡建设	0	县丁字街的商户乱	县 丁字街 商户 摆摊					
7	5	K8县南门街	城乡建设	0	县南门街干净整洁	县 南门 街 干净 整洁 几天 样子					
8	6	K8县冷江东	城乡建设	0	县冷江东路蓝波旺	县冷 江东 蓝波 旺 酒店 外墙 装修 无人 施工					
9	7	K8县九亿广	城乡建设	0	县九亿广场的公厕	县 九亿 广场 公厕 安装 照明灯					
10	8	K4县石期市	城乡建设	0	县石期市镇老农贸	县 石期 市镇 农贸市场 公厕 旱厕					
11	9	K市域轨道	城乡建设	0	市域轨道交通规划	市域 轨道交通 规划 建议					
12	10	关于K市域	城乡建设	0	关于市域轨道交通	市域 轨道交通 规划 建议					

图 5 部分留言数据及其分词效果

### 2.1.3 TF-IDF 文本特征选择

在文本分类中，当分类文本数目比较大的时候，从文本中提取的特征也就随之增加，但有些特征对文本的分类作用并不大。所以我们要应用适当的文本特征选择方法来提取具有代表性的特征<sup>[2]</sup>。本题采用基于 TF-IDF 的文本特征提取。

TF-IDF (term frequency - inverse document frequency) 是一种用于信息检索与数据挖掘的常用加权技术。TF-IDF 是在单词计数的基础上，降低了常用高频词的权重，增加罕见词的权重。因为罕见词更能表达文章的主题思想。

算法步骤：

TF-IDF 的计算方法实际上是词频与逆文档频率的乘积，即  $TF * IDF$ 。词频是词  $t$  在文档  $d$  中出现的频率，而逆文档频率代表了词  $t$  的类别区分能力，包含词  $t$  的文本越少则  $IDF$  越大。某个词对文本的重要性越高，它的 TF-IDF 值就越大。

TF 和 IDF 的计算公式分别如 (1) 和 (2) 所示：

$$tf(t, d) = \frac{f(t, d)}{\sum_k f(w_k, d)} \quad (1)$$

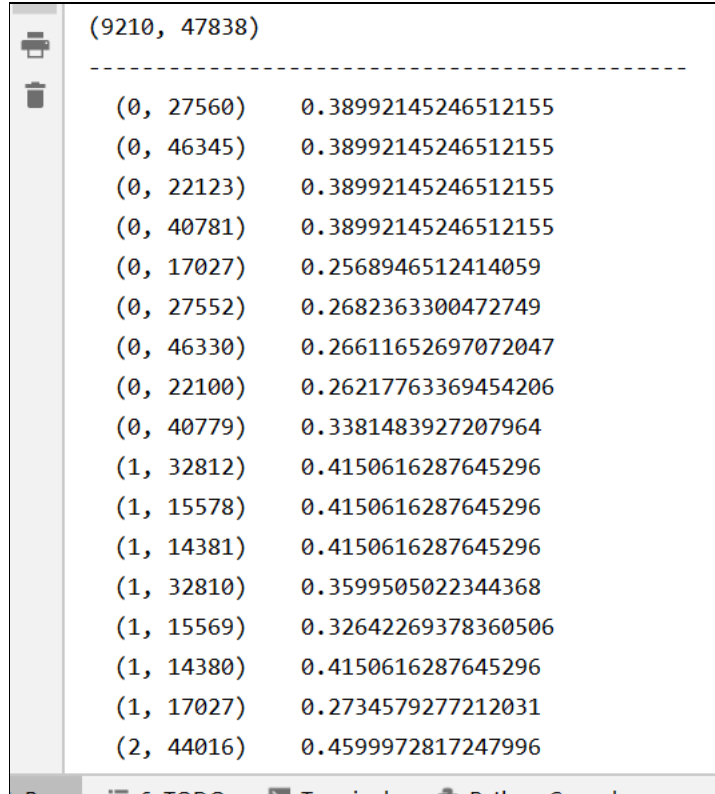
$$idf_t = \log\left(\frac{N}{1 + df_t}\right) \quad (2)$$

因此接下来需要计算 cut\_content 字段的 TF-IDF 的特征值。

本题使用 `sklearn.feature_extraction.text.TfidfVectorizer` 方法来抽取文本的 TF-IDF 的特征值。这里我们使用了参数 `gram_range=(1, 2)`，这表示我们除了抽取评论中的每个词语外，还要抽取每个词相邻的词并组成一个“词语对”，如：词 1，词 2，词 3，词 4，(词 1，词 2)，(词 2，词 3)，(词 3，词 4)。这样就扩展了我们特征集的数量，有了丰富的特征集才有可能提高我们分类文本的准确度。参数 `norm='l2'`，是一种数据标准划处理的方式。提取特征值的程序段如下所示：

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(norm='l2', gram_range=(1, 2))
features = tfidf.fit_transform(df.cut_content)
labels = df.label_id
```

如图 6 所示，我们看到我们的 features 的维度是(9210,47838),这里的 9210 表示我们总共有 9210 条留言数据，47838 表示数据的特征数量，这包括全部留言数据中的所有词语数+词语对(相邻两个单词的组合)的总数。



(9210, 47838)	
(0, 27560)	0.38992145246512155
(0, 46345)	0.38992145246512155
(0, 22123)	0.38992145246512155
(0, 40781)	0.38992145246512155
(0, 17027)	0.2568946512414059
(0, 27552)	0.2682363300472749
(0, 46330)	0.26611652697072047
(0, 22100)	0.26217763369454206
(0, 40779)	0.3381483927207964
(1, 32812)	0.4150616287645296
(1, 15578)	0.4150616287645296
(1, 14381)	0.4150616287645296
(1, 32810)	0.3599505022344368
(1, 15569)	0.32642269378360506
(1, 14380)	0.4150616287645296
(1, 17027)	0.2734579277212031
(2, 44016)	0.4599972817247996

图 6 TF-IDF 文本特征选择的部分结果

## 2.1.4 支持向量机（LinearSVC）

群众留言主题内容的自动分类属于典型的多类分类问题，经典的 SVM 算法是解决二类分类问题，因此需要将算法扩展到多类分类模型。常见的解决思路是将多类问题分解为多个两类问题求解，通过决策函数确定分类结果。采用 Python3.6 调用 sklearn 库中的 svm 包导入 LinearSVC，实现多类分类算法，其基本思想是找出能使得不同类别样本数据的分类间隔最大的超平面，本题采用 LinearSVC 进行文本分类的具体过程如下。

将文本特征选择之后合成的矩阵作为支持向量机的输入，通过训练集训练模型，测试集评估分类效果。因此，将数据初步按照 test\_size=0.33 的比例划分成训练集、测试集和验证集，采用 svm.LinearSVC 训练模型，模型的训练过程即寻找支持向量，确定最大超平面的过程<sup>[3]</sup>。

定义含有  $n$  个样本的训练集合为  $T=\{(x_i, y_i)\}$ ， $x_i=(x_1, x_2, \dots, x_n)$ ，类别  $y_i \in \{+1, -1\}$ ， $n$  维空间中线性判别函数的一般形式为  $g(x)=w^T x+b$ ，分类的超平面方程表示为  $w^T x+b=0$ ， $\text{margin}(b,w)$  表示超平面  $w^T x+b$  离样本点的最小距离，而

优化问题的目标则是让这个最小距离最大化。除了速度上的优势外，LinearSVC 模型还能更灵活地选择正则项和损失函数。默认的正则项是 L2 范数，损失函数 loss 默认为 squared\_hinge，C 值为 1，最多迭代次数 max\_iter 默认为 1000 次。在进行多分类时，multi\_class 默认值为 ovr(one-vs-rest)。由于本题中各类别不平衡，因此可以使用 class\_weight 和 sample\_weight 参数进行设置。如果不指定该参数，各类的权重均为 1；如果该参数设置为“balanced”模式，可以使得损失函数对样本数不足的数据更加关注，模型将根据输入的类别标签 y，自动将各类的权重设置为  $n\_samples/(n\_classes * np.bincount(y))$ ，其中 np.bincount() 用来统计 array 中各个值出现的次数，尤其适合统计类别标签。但需要注意的是，LinearSVC 不接受关键词 kernel，因为它被假设是线性的。

支持向量分类的优势在于可以筛选出对预测任务最为有效的少数样本，节省模型学习所需要的数据内存，同时也提高了模型的预测性能。

## 2.1.5 评价指标

评价指标，我们一般最先想到的是分类准确率(Accuracy)，但准确率并不是全部。如果一个分类模型的准确率很高而查全率很低(Recall)，那么反而意味着模型没能把其他本应该预测出来的类别给预测出来。

表 2 是两分类器混淆矩阵，其中 TP 表示实际为正类、预测也为正类的样本数量；FN 表示实际为正类、预测为反类的样本数量；FP 表示实际为反类、预测为正类的样本数量；TN 表示实际为反类、预测也为反类的样本数量<sup>[4]</sup>。

表 2 两分类混淆矩阵

	预测正例	预测反例
实际正例	TP	FN
实际反例	FP	TN

本题使用查准率(Precision)、查全率(Recall)、F-Score 和准确率(accuracy)对分类方法进行评价，它们分别定义（3）至（6）的公式：

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$F_1 = \frac{1}{n} \sum_{i=1}^n \frac{2P_i R_i}{P_i + R_i} \quad (5)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

其中 F1 公式中  $P_i$  为第  $i$  类的查准率， $R_i$  为第  $i$  类的查全率。F1-Score 指标综合了 Precision 与 Recall 的产出的结果。F1-Score 的取值范围从 0 到 1 的，1 代表模型的输出最好，0 代表模型的输出结果最差。

## 2.2 问题 2 分析方法与过程

在进行操作之前要明确具体步骤，以下是对问题 2 的操作流程，如图 7 所示：

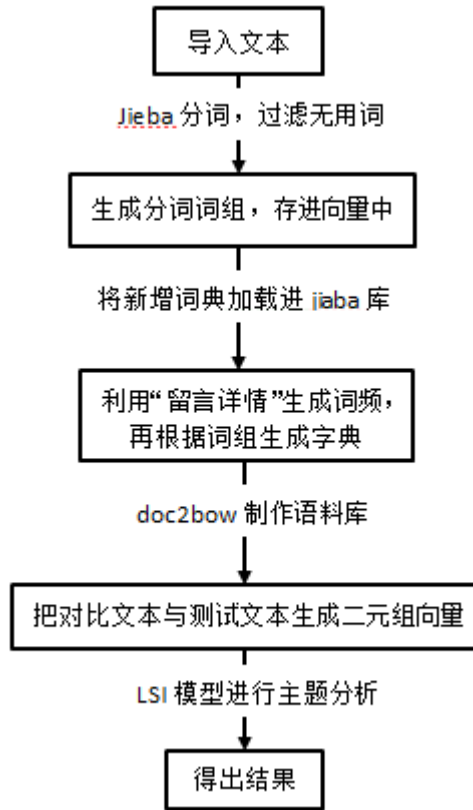


图 7 问题 2 操作流程

### 2.2.1 数据的选取

主要详细介绍本文数据来源及最后目的。

本文选取了收集自互联网公开来源的群众问政留言记录。对数据处理是先读取示例数据中的附件 3，根据表中“留言主题”及“留言详情”内容，主要可将

某一时段内群众集中反映的问题分为一下 15 个热点，其中前五个主要为环境污染加上扰民、学校相关事务、医疗服务及消费、建筑协商、车辆及交通管理等一系列相关投诉建议等问题。

### 2.2.2 自定义词典与词性

在进行中文分词过程中，通常会遇到一些专用词语无法精准的切分，虽然 jieba 工具有新词识别能力，但也无法识别出所有 jieba 词库里没有的词语，但它为开发者提供了添加自定义词典功能，从而保证更好的分词正确率。其函数原型如下：`load_userdict(f)`。该函数只有一个参数，表示载入的自定义词典路径，`f` 为文件类对象或自定义词典路径下的文件。词典的格式为：一个词占一行，每行分为三部分：`word freq word_type`。其中，`word` 为对应的词语，`freq` 为词频（可省略），`word_type` 为词性（可省了），中间用空格隔开，顺序不可以改变。注意，文件必须为 UTF-8 编码。

问题 2 利用建立模型求相似度得出最终结果。为提高文本的相似度，使用自定义 `new.txt` 文本词典添加到 jieba 库中。使用的是 `f` 为自定义词典路径下的文件，词典的格式只有 `word` 部分。当利用 `jieba.cut` 时，就不会误判词组，使文本更加简洁。

### 2.2.3 分词及去除停用词

（1）jieba 库中的三种分词模式：

- 1) 精确模式，试图将句子最精确地切开，适合文本分析；
- 2) 全模式，把句子中所有的可以成词的词语都扫描出来,速度非常快，但是不能解决歧义；
- 3) 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

由于本文出现大量的敬辞，为提高相似度在分词前先把这些词去掉，利用 `re` 中的 `apply()` 函数作相关处理，得到初步文本。再引用全模式中的 `jieba.lcut(s)` 函数，将初步文本切分，返回的是一个列表类型的词组表。

（2）停用词

停用词是指在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为 Stop Words（停用词）。这些停用词都是人工输入、非自动化生成的，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。

文本在分词之后，对得到的词组列表进行去除停用词步骤，得到有效词组列表。经过处理后的文本会更加简洁清晰，所得的匹配度也会更高，在之后的文本相似度对比时能更精确。下个步骤即使将有效词组列表保存在 all\_doc\_list 列表中，定义测试文本，并将测试文本保存在 doc\_test\_list 列表中。

#### 2.2.4 词频计算

词频可以反映出总体内容的关键词，可以通过获取到词频最高的关键词来反应热点，将此处词频的统计加上后面训练主题模型结合起来，可明确附件 3 中的主要的热点问题。处理本文的词频，是使用 TextRank 算法原理，该算法的介绍如下。

用 TextRank 提取来提取关键词，用 PageRank 的思想来解释它：如果一个单词出现在很多单词后面的话，那么说明这个单词比较重要。一个 TextRank 值很高的单词后面跟着的一个单词，那么这个单词的 TextRank 值会相应地因此而提高，这样 TextRank 的公式就可以由 PageRank 公式改写为：

$$S(v_i) = (1 - d) + d \sum_{(j,i) \in \varepsilon} \frac{w_{ji}}{\sum_{v_k \in out(v_j)} w_{jk}} S(v_j)$$

公式的意思为，TextRank 中一个单词 i 的权重取决于与在 i 前面的各个点组成的 (j, i) 这条边的权重，以及 j 这个点到其他边的权重之和。经过统计分析可得出最终词频。经过处理后会得到一个词频表，部分结果如图 8 所示：



	A	B	
1	word	count	
2	小区	4987	
3	业主	4956	
4	A市	4519	
5	领导	3248	
6	部门	2884	
7	政府	2776	
8	相关	2756	
9	居民	2309	
0	希望	2286	
1	开发商	2247	
2	公司	2001	
3	影响	1913	
4	解决	1865	
5	情况	1864	
6	物业	1787	
7	生活	1560	
8	社区	1512	
9	区	1456	
0	西地省	1416	
1	街道	1408	
2	学校	1386	

图 8 词频统计部分结果

## 2.2.5 制作语料库

语料库制作是为了计算相似度而制作。在制作语料库前，先用 `dictionary` 方法获取词袋 (`bag-of-words`)，词袋中用数字对所有文本中的词进行编号，编号与所有词之间有一一对应关系。词袋模型被广泛应用在文件分类，词出现的频率可以用来当作训练分类器的特征。词袋模型就是把一篇文本想象成一个个词构成的，所有词放入一个袋子里，没有先后顺序、没有语义。词袋模型下，像是句子或是文件这样的文字可以用一个袋子装着这些词的方式表现，这种表现方式不考虑文法以及词的顺序。最近词袋模型也被应用在电脑视觉领域<sup>[5]</sup>。

将列表中所有的词利用词袋编好号后，接着使用 `doc2bow` 制作语料库，指经科学取样和加工的大规模电子文本库。借助计算机分析工具，研究者可开展相关的语言理论及应用研究。语料库是语料库语言学研究的基础资源，也是经验主义语言研究方法的主要资源。应用于词典编纂，语言教学，传统语言研究，自然语言处理中基于统计或实例的研究等方面。语料库是一组向量，向量中的元素是一个二元组（编号、频次数），对应分词后的文本中的每一个词，把 `all_doc_list`

和 `doc_test_list` 列表中的词转换为二元组的向量，把这个二元向量组的向量保存在 `doc_test_vec` 中。

## 2.2.6 建立 LSI 模型

### LSI 模型

TF-IDF 模型足够胜任普通的文本分析任务，用 TF-IDF 模型计算文本相似度已经比较靠谱了，但是细究的话还存在不足之处。实际的中文文本，用 TF-IDF 表示的向量维数可能是几百、几千，不易分析计算。此外，一些文本的主题或者说中心思想，并不能很好地通过文本中的词来表示，能真正概括这篇文本内容的词可能没有直接出现在文本中。

因此，这里引入了 Latent Semantic Indexing (LSI) 从文本潜在的主题来进行分析。Latent Semantic Indexing-隐性语义索引，也可译为隐含语义索引，是近年来逐渐兴起的不同于关键词检索的搜索引擎解决方案，其检索结果的实际效果更接近于人的自然语言，在一定程度上提高检索结果的相关性，目前已被逐渐的应用到图书馆、数据库和搜索引擎的算法当中。Google 就是典型的代表。LSI 是概率主题模型的一种，核心思想是：每篇文本中有多个概率分布不同的主题；每个主题中都包含所有已知词，但是这些词在不同主题中的概率分布不同。LSI 通过奇异值分解的方法计算出文本中各个主题的概率分布，严格的数学证明需要看相关论文。假设有 7 个主题，那么通过 LSI 模型，文本向量就可以降到 15 维，每个分量表示对应主题的权重。

## 2.2.7 相似度分析

使用 LSI 模型对语料库建模，获取测试文本中，每个词的值，对每个目标文本，分析与测试文本的相似度，根据相似度从高到低排序，相似度从高到低排序，得出最终结果。

## 2.3 问题 3 分析方法与过程

### 2.3.1 问题 3 流程图

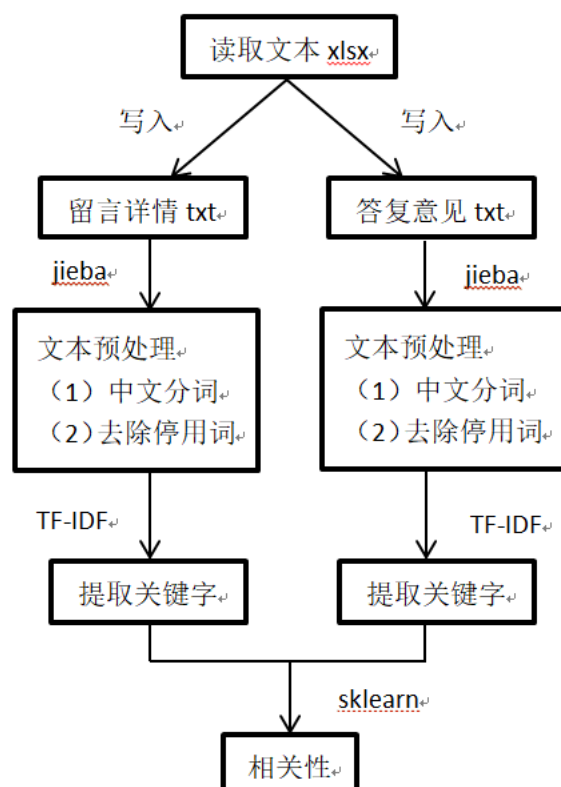


图 9 问题 3 操作流程

### 2.3.2 文本预处理

#### (1) 对文本进行中文分词

在对这些文本进行应用之前，要对其进行文本预处理。我们利用 python 的中文分词包 jieba 对文本进行中文分词。jieba<sup>[6]</sup>基于 Trie 树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）。采用了动态规划查找最大概率路径，找出基于词频的最大切分组合。对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。从源代码的角度分为三部分对 jieba 中文分词进行分析：

1) jieba 分词的初始化，包括核心词典和用户词典的加载，这一部分涉及最



明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。

### 2.3.3 提取关键词<sup>[8]</sup>

将文本预处理好了之后，我们要提取本文中的关键字。如果某个词很重要，它应该在文本中多次出现。于是，我们进行“词频”（Term Frequency，缩写为 TF）统计。在词频的基础上，要对每个词分配一个“重要性”权重。这个权重叫做“逆文档频率”（Inverse Document Frequency，缩写为 IDF），它的大小与一个词的常见程度成反比。将“词频”（TF）和“逆文档频率”这两个值相乘，就得到了一个词的 TF-IDF 值。某个词对文本的重要性越高，它的 TF-IDF 值就越大，所以，排在最前面的几个词，就是该文本的关键词。

### 2.3.4 TF-IDF 算法<sup>[9]</sup>

在对文本信息分词后，需要把这些词语转换为向量，以供挖掘分析使用。关键词的提取，采用 TF-IDF 算法，把文本信息转换为权重向量。TF-IDF 算法的具体原理如下：

第一步，计算词频：

$$\text{词频(TF)} = \text{某个词在文章中的出现次数}$$

考虑到文本有长短之分，为了便于不同文章的比较，进行“词频”标准化。

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

或者

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{该文出现次数最多的词的出现次数}}$$

第二步，计算逆文档频率，这时，需要一个语料库（corpus），用来模拟语言的使用环境：

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1}\right)$$

如果一个词越常见，那么分母就越大，逆文档频率就越小越接近 0。分母之所以要加 1，是为了避免分母为 0（即所有文档都不包含该词）。log 表示对得到的值取对数。

第三步，计算 TF-IDF：

$$\text{TF-IDF} = \text{词频(TF)} \times \text{逆文档频率 (IDF)}$$

可以看到，TF-IDF 值与一个词在文档中的出现次数成正比，与该词在整个语言环境中的出现次数成反比。所以，自动提取关键词的算法就很清楚了，就是计算出文档的每个词的 TF-IDF 值，然后按降序排列，取排在最前面的几个词。

### 2.3.5 关键字相关性

通过 TF-IDF 算法，从留言详情和答复意见提取到的关键词，将留言详情的关键词合并成一个集合，计算答复意见对于这个集合中的词的词频（为了避免文本长度的差异，可以使用相对词频），利用使用 `from sklearn.metrics.pairwise import cosine_similarity` 对两者关键词之间做相关性矩阵，使用的是余弦相似度计算，值越大就表示越相似。

## 3.结果分析

### 3.1 问题 1 结果分析

#### 3.1.1 分类结果评估

accuracy	0.85	3040
----------	------	------

图 11 分类结果准确率

根据评估结果，见图 11，总体分类结果准确率为 85%，表明本次分类比较合理，效果较好。但是多分类模型一般不使用准确率(accuracy)来评估模型的质量，因为本题的训练数据不平衡，accuracy 不能反映出模型的实际预测精度，这时候我们就需要借助于 F1 分数等指标来评估模型。下面我们查看各个类的 F1 分数。

表 3 Linear 分类结果

类别	Precision(%)	Recall(%)	F <sub>1</sub> -score(%)
城乡建设	0.77	0.86	0.81
环境保护	0.89	0.86	0.87
交通运输	0.87	0.84	0.85
教育文体	0.91	0.87	0.89
劳动和社会保障	0.84	0.92	0.88
商贸旅游	0.88	0.73	0.80
卫生计生	0.86	0.82	0.84

此次实验运用的计算机处理器型号为 Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80GHz，内存为 8.00GB，训练数据集，将剩余预测集进行分类并评估结果的过程共耗时 17.4032355 秒。在此预测集中共有 2582 个群众留言数据归纳入了原有选项当中，约占所有预测集数据的 84.93%。

通过对分类结果的查准率、查全率率和 F1-score 进行统计（见表 3），可以得到如下实验结论：利用文本挖掘技术以及支持向量机分类器，对留言信息进行深入剖析和提取，从而能快速实现类别分类，在性能和正确率上都远远优于传统

方法。且相较于人工分类，采用文本自动分类可以大大节省时间，并可从原始数据中许多群众留言数据结果重新归类，从而提高数据质量。



图 12 混淆矩阵

针对 LinearSVC 模型，我们查看混淆矩阵，并显示预测标签和实际标签之间的差异。混淆矩阵的主对角线表示预测正确的数量，除主对角线外其余都是预测错误的数量。从图 12 所示的混淆矩阵可以看出“商贸旅游”预测的错误数量较多。经分析存在差异的原因是样本数据存在一定程度的不均衡，城乡建设类的数量最多，导致城乡建设类的支持向量过多，从而使得超平面偏向数据少的一侧，导致其他类别的准确率会偏低。

### 3.2 问题 2 结果分析

#### 3.2.1 对热点问题表分析

通过利用 Python 工具，对附件 3 的数据进行处理，整理得到结果如图 13 所示：



A	B	C	D	E
热度排名	问题ID	时间范围	地点人群	问题描述
1	718	2017/06/08至2019/11/12	学校师生	学校管理及学生意见
2	703	2019/01/06至2020/01/02	A市群众	环境加上噪声污染严重扰民
3	638	2019/01/01至2019/12/31	A市居民	施工隐患
4	589	2019/01/06至2020/01/02	市区出行者	车辆带来的生活困扰
5	543	2019/01/02至2019/12/31	医院医患人员	医疗服务及资源问题

图 13 热点问题

由于留言的问题中会出现多个内容的情况，即可将一个问题归为多类，所以在归类问题中也会出现问题重复使用的情况。而热度排名和问题 ID 是通过结果分析中，一类问题相似度匹配最多的来排名的，相似度高的分为一类，并归为同一个问题 ID，热度指数来源于一类问题中占有居民反应的总数量，该数即为热度指数；人群或者地点是通过留言主题中提取，既方便又易取，问题描述是利用之前统计的词频来获取的，在词频中可以反应出最热门的信息，所以利用词频组成问题描述可以减少工作量。时间范围是在这一类问题从最早开始出现反应至今的时间，即为时间范围，对时间的格式进行处理，输出结果如图 14 所示：

```

0      2019/2/28
1      2019/2/14
2      2019/7/19
3      2019/8/19
4      2019/11/22
...
4321   2019-11-22
4322   2019-11-05
4323   2019-04-28
4324   2018-05-17
4325   2017-06-08

```

图 14 对时间处理的结果

在设置主题时，经过多次训练及附件 1 中的问题分类，再结合附件 3 可大致分为 15 个主题，可以利用这几个主题测试与之匹配的各个文本，通过以上各种方式对问题处理后，可得出各个居民反应的文本相似度，经过模型训练得出的 15 个主题结果如图 15 所示：

```

(1, '0.470*"业主" + -0.434*"A市" + -0.262*"公司" + 0.249*"小区" + -0.190*"区" + -0.166*"学校" + 0.148*"物业" + -0.133*"中学" + -0.112*"号" + -0.109*"教育"'),
(2, '0.464*"买受人" + -0.363*"小区" + 0.283*"出卖" + 0.203*"条款" + 0.198*"房屋" + 0.191*"合同" + 0.150*"装修" + 0.134*"开发商" + 0.129*"商品房" + -0.119*"区"'),
(3, '-0.620*"公司" + 0.271*"学校" + 0.236*"中学" + 0.196*"教育" + 0.184*"区" + 0.129*"小学" + 0.123*"教育局" + -0.117*"有限公司" + -0.116*"请" + 0.096*"市属"'),
(4, '-0.554*"小区" + -0.350*"买受人" + 0.299*"开发商" + 0.294*"业主" + -0.216*"出卖" + -0.177*"居民" + -0.137*"社区" + 0.113*"A市" + -0.112*"医院" + -0.106*"条款"'),
(5, '0.368*"A市" + -0.360*"公司" + -0.321*"学校" + -0.237*"中学" + -0.201*"教育" + 0.186*"商品房" + -0.148*"业主" + -0.131*"教育局" + -0.129*"买受人" + 0.117*"开发商"'),
(6, '-0.348*"小区" + 0.251*"社区" + 0.209*"区" + 0.207*"物业" + -0.186*"开发商" + -0.181*"商品房" + 0.157*"餐饮" + -0.154*"公司" + 0.143*"部门" + -0.140*"销售"'),
(7, '-0.463*"小学" + -0.280*"砂子" + -0.270*"塘" + -0.254*"校区" + -0.209*"A5" + 0.209*"中学" + -0.162*"雅苑" + -0.160*"建发决玺" + -0.156*"路" + -0.154*"楼盘"'),
(8, '0.306*"A市" + 0.230*"区" + -0.193*"餐饮" + 0.193*"物业" + -0.177*"县" + -0.169*"A8" + -0.169*"部门" + -0.158*"栋" + -0.156*"学校" + 0.132*"A4"'),
(9, '0.311*"路" + -0.233*"餐饮" + -0.215*"商品房" + -0.192*"小学" + -0.158*"经营" + -0.156*"服务" + -0.134*"部门" + 0.133*"A9" + -0.129*"销售" + 0.123*"规划"'),
(10, '-0.415*"社区" + -0.241*"尖山" + 0.197*"区" + -0.196*"装修" + 0.190*"路" + -0.149*"政府" + -0.148*"A8" + -0.140*"社员" + 0.132*"餐饮" + -0.132*"县"'),
(11, '-0.378*"装修" + 0.337*"A8" + 0.326*"县" + 0.193*"学校" + -0.172*"全" + -0.153*"中学" + 0.151*"A市" + -0.137*"区" + -0.123*"公示" + -0.122*"消费者"'),
(12, '0.212*"A9" + -0.208*"区" + -0.194*"开发商" + 0.193*"垃圾" + 0.191*"市" + -0.171*"A4" + 0.171*"城市" + -0.170*"说" + 0.161*"社区" + -0.155*"政府"'),
(13, '0.331*"装修" + -0.289*"社区" + -0.266*"开发商" + 0.243*"物业" + -0.207*"尖山" + 0.205*"A8" + 0.190*"县" + 0.164*"全" + -0.139*"中学" + -0.120*"社员"'),
(14, '-0.260*"社区" + -0.245*"路" + -0.184*"物业" + -0.180*"说" + 0.178*"土地" + -0.155*"尖山" + 0.150*"村民" + 0.149*"建筑" + -0.139*"可以" + 0.130*"A4"')]]

```

图 15 “留言详情”训练主题结果

由运行结果可知，训练得出来的热点问题，前五个主要就是教育、医疗、扰民、餐饮、交通等问题。经过以上的热点分类，可将附件 3 中 4327 条问题进行归类整理，反馈给相关管理部门处理，可极好的提高办事效率。同时也会出现一问多主题的现象，所以在分析热点问题上会出现一个问题可归为多类的。

最后经过整理可输出前五个热点问题的柱状结果如图 16 所示：

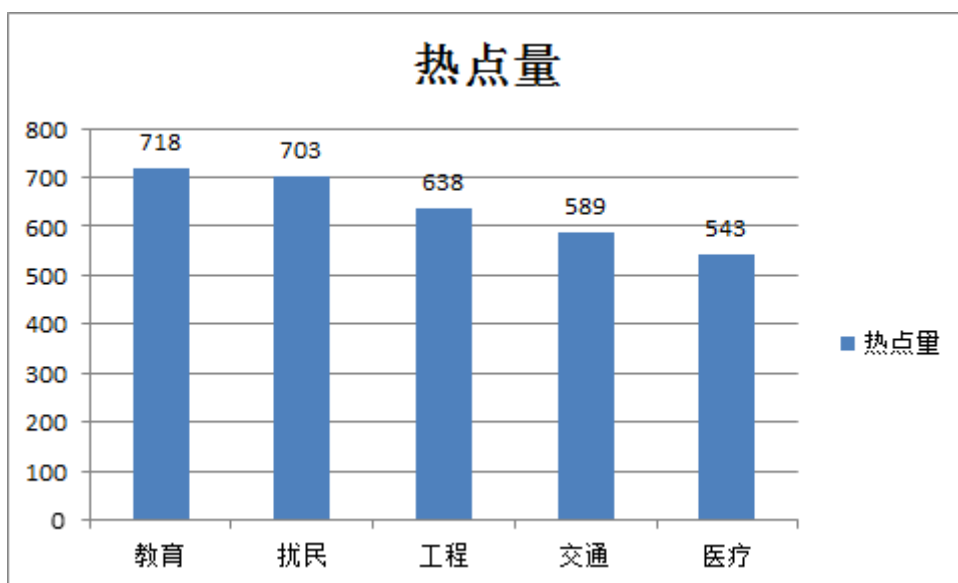


图 16 前五热点柱状图

### 3.2.2 对热点问题留言明细表分析

本文主要有 4236 条数据，现主要是利用以上方法对这些数据进行处理，归类后进行数量统计，得出最终热点排名，由上可知排名前五的热点问题，下面将继续对热点问题进行分类整理，并将以上前五的热点主要内容显示出来。通过训练模型及词频排名等各种步骤可将附件 3 分为 15 类，即下面将对这些以排好名的数据显示在热点问题分析表中。其中问题 ID 是通过同类问题反馈的数量排序得出的结果，热点分析表主要问题如表 4 所示：

表 4 热点 ID 结果

问题 ID	主要问题	热点指数
1	教育	718
2	环境卫生	703
3	建设建筑	638
4	交通	589
5	医疗卫生	543
6	农业	511
7	法治政治	198
8	业余营业	164

9	住房条款	138
10	科技信息化	47
11	劳动人民维权	32
12	商业资源	30
13	民生民权	21
14	服务管理	15
15	福利	10

根据热度表 4 的结果，可得出热点指数占总比例的分布值，具体分析结果如图 17 所示：

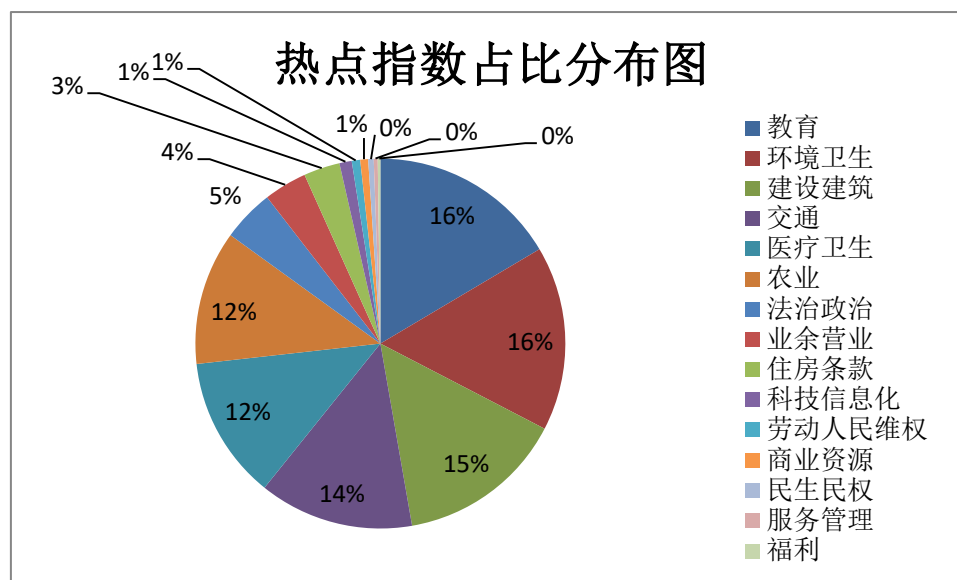


图 17 热点指数占比分布图

### 3.3 问题 3 结果分析

#### 3.3.1 相关性分析

数据中的附件 4，一共有 2816 条留言详情，以及 2816 条答复意见，分别取出每一条留言详情和答复意见的关键字，对这两者的关键字做相关性处理，这两者的相关性矩阵结果如图 18 所示：

	0	1	2	3	4	5	...	2810	2811	2812	2813	2814	2815
0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2811	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0
2812	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0
2813	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0
2814	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0
2815	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0

图 18 相关性矩阵结果图

有相关性矩阵可知，该矩阵是一个 2816\*2816 的单位矩阵，由此说明每一条留言详情的关键字与答复意见的关键字的相关性程度极强。

### 3.3.2 完整性分析

关于答复意见数据的完整性，首先说说它的含义，数据完整性是指数据的正确性、一致性和有效性,是指数据中不应该存在不符合语义的数据。我们首先对留言详情和答复意见用 jieba.cut 进行分词。然后用 stopwordslist 创建停用词表，并去除停用词。随后，使用 TF-idf 词袋模型，对特征进行向量化数字映射。最后，使用 `from sklearn.metrics.pairwise import cosine_similarity`，对这两组关键字之间做相关性矩阵，使用的是余弦相似度计算公式。所以，该答复意见具有完整性。

### 3.3.3 可解释性分析

广义上的可解释性指在我们需要了解或解决一件事情的时候，我们可以获得我们所需要的足够的可以理解的信息。

建模之前的可解释性方法，主要涉及数据的预处理，这里主要用到的数据预处理是中文分词和去除停用词，去除停用词是因为在数据中，有大量的标点符号，以及语气词，在提取关键字的时候，如果提取到的是这些，是没有用的，所以要去除停用词。

建立本身具备可解释性的模型，在这里就是自然语言处理和文本挖掘的方法

解决问题。在这里，我们的思路是利用关键字，根据 `from sklearn.metrics.pairwise import cosine_similarity`，对这两组关键字之间做相关性矩阵，用是余弦相似度计算公式，计算出留言详情和答复意见的相关性。

建模之后，作出解释，由 3.3.1 的相关性分析，以及相关矩阵运行结果图可知，该模型设想正确。

## 4.结论

问题 1 对基于支持向量机的群众留言文本进行了文本分类实验。对不同类别的数据计算各自的查准率，查全率和 F1-Score 值，分析了各个类别的分类结果。从问题 1 中可以看出，运用文本挖掘技术处理网络问政平台的群众留言具有很高的效率和较高的准确率，将其运用网络问政平台的群众留言结果分析中可以大大降低人力、财物投入，优化了现有的人工审核处理流程，提高了职能部门处理群众留言的效率，也能有效获取人民群众的真实意见建议，能更好地服务于党政机关相关决策，具备一定的合理性和科学性。但问题 1 也有一些不足之处，一是缺乏已留言数据类别对应的具体职能部门数据，因此匹配环节未能具体实现以验证匹配效果，二是未进行其他分类算法的效果对比。

问题 2 主要是利用 LIS 模型训练得出主题，加上词频结果可以更加直观的看出文本热点。热点问题提取的好处可以有效降低工作量提高工作效率，虽目前 LIS 模型的使用已经不再广泛，但仍有其特色，它能够通过提取文本的隐含主题来匹配其他内容，所以还是非常好用的。在词频的统计上，非常有效明确。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。

问题 3 主要是相关部门对留言详情作出答复意见的分析，因各类社情民意相关的文本数据量不断攀升，给相关部门的工作带来了极大挑战，所以利用 jieba.cut 进行分词，并去除停用词，对文本进行预处理，随后提取关键字，最后，根据 `from sklearn.metrics.pairwise import cosine_similarity`。对这两组关键字之间做相关性矩阵，用是余弦相似度计算公式，计算出留言详情和答复意见的相关性。从答复意见的相关性、完整性、可解释性等角度进行分析，并结合相关性矩阵结果可以看出答复意见的质量较高。

## 5.参考文献

- [1] 陈曦.文本挖掘技术在社情民意调查中的应用[J].中国统计.2019.
- [2] 张玉芳,彭时名,吕佳.基于文本分类 TFIDF 方法的改进与应用.计算机工程.2006,32(19): 76-78.
- [3] 王志明,沙莎.Web 文本挖掘技术在新闻主题检测中的应用研究[J].长沙大学学报.2017.
- [4] 段尧清,姚兰政.媒融合问政平台非正式文本自动分类匹配研究[J].情报理论.2020.
- [5] 王涛.基于词袋模型的人脸表情识别研究[D].华中科技大学.2013.
- [6] 曾小芹.基于 Python 的中文结巴分词技术实现[J].豫章师范学院数学与计算机学院.2019.
- [7] 马治涛.文本分类停用词处理和特征选择技术研究[J].西安电子科技大学.2014.
- [8] 孙士保,李保元,李天瑞,吴正江,郑瑞娟.基于类内关键词的中文文本分类模型的改进[J].广西师范大学学报(自然科学版).2009(03).
- [9] 宋章浩.中文文本分类中 TF-IDF 方法的改进与应用[J].科技展望.2014(22).