“智慧政务”中的文本挖掘应用

摘要：

随着网络问政平台的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势。本文利用**朴素贝叶斯算法**、**LDA 模型**以及 **F-score 评价**，对群众留言分类、热点问题挖掘、答复意见评价这三个问题进行建模，利用 Python 对数据进行处理得到结果。此外本文对数据进行**仿真性模拟**发现效果良好。

针对问题 1，我们首先对数据进行预处理。根据附件一，将留言分为以下七类：城乡建设、环境保护、交替运输、教育文体、劳动和社会保障、商贸旅游以及卫生计生。对每一类数据进行汇总以及筛选。其次根据**朴素贝叶斯算法**建立一级**标签分类模型**，根据留言主题与留言详情选取特征词，利用 **Python** 进行编程，得到在该模型下的**查全率**为：**0.9600558**，**查准率**为：**0.9782507**。其次利用 F-score 对该模型进行评价，所得结果为：**0.9684174**。根据结果可知，朴素贝叶斯算法所建立的一级标签分类模型良好。

针对问题 2，首先对附件 3 中的数据进行筛选，剔除有冲突以及矛盾数据，将数据通过留言主题以及留言详情按照区域进行划分，然后根据问题 1 中构建的一级标签分类模型对留言进行分类处理。其次根据附件 3 中数据我们选取起止时间作为时间段，利用 **LDA 模型**以及 **DBScan 聚类算法**，根据留言数量、点赞数、留言持续时间这三个热度评价指标，构建热度值评价模型。利用**吉布斯采样**估计 LDA 模型中的参数，得出基于时间消退的热度值计算方法。最后利用算法进行遍历，筛选出热度较高的留言问题：小区附近搅拌站噪音扰民，污染严重。

针对问题 3，我们主要考虑留言以及答复意见之间的相关性程度。首先基于文本数据，我们利用特征词来进行建立相关性分析，通过 **JS 距离**计算留言和答复意见的相关性程度。其次根据答复的相关性、完整性、可解释性、以及留言答复间隔时间作为评价指标，建立答复意见的质量**评价指标体系**，其中我们根据**最大熵模型**来确定每项评价指标的权重，最后根据计算结果为：7.8536，我们可以推断出答复意见的质量良好。

本文所涉及算法较为完整地处理了附件所给数据且去掉了足够多的矛盾数据，其中评价指标是一步一步来构建的，并且对于一些小细节的处理足够精致，并且在计算过程中算法源程序都是正确的。

关键词：**朴素贝叶斯算法 LDA 模型 DBScan 聚类算法 最大熵模型 评价指标体系**

Text Mining Application in "Smart Government Affairs"

Abstract:

With the development of the network questioning platform, the establishment of a smart government system based on natural language processing technology has become a new trend in the development of social governance innovation. This paper uses the **Naive Bayes algorithm, LDA model and F-score evaluation** to model the three questions of people's message classification, hot topic mining, and answer evaluation, and uses **Python** to process the data to obtain the results. In addition, this paper makes a good simulation of the data and finds that the effect is good.

For problem 1, we first preprocess the data. According to Annex I, the message is divided into the following seven categories: urban and rural construction, environmental protection, alternate transportation, education and culture, labor and social security, trade and tourism, and health and family planning. Summarize and filter each type of data. Secondly, a first-level label classification model is established according to the **Naive Bayes algorithm**, feature words are selected according to the subject and details of the message, and programming is performed using Python. The **recall rate** under this model is 0.9600558 and the **precision rate is 0.9782507**. Secondly, the model is evaluated by **F-score**, and the result is 0.9684174. According to the results, the first-level label classification model established by the Naive Bayes algorithm is good.

For question 2, first filter the data in Annex 3, remove conflicting and contradictory data, divide the data by message subject and message details by region, and then classify the messages according to the first-level **label classification model** constructed in question 1. deal with. Secondly, according to the data in Annex 3, we select the start and end time as the time period, and use **the LDA model and DBScan clustering algorithm** to build a heat value evaluation model based on the three heat evaluation indicators of the number of comments, the number of likes, and the duration of the message. Using **Gibbs sampling** to estimate the parameters in the LDA model, the calculation method of the heat value based on the time fading is obtained. Finally, the algorithm is used to traverse, and the hot message problem is selected: the noise of the mixing station near the community disturbs the people and the pollution is serious.

For question 3, we mainly consider the degree of relevance between the message and the answer. First of all, based on text data, we use feature words to establish correlation analysis, and calculate the degree of relevance of message and reply opinions through **JS distance**. Secondly, based on the relevance, completeness, interpretability of the reply, and the interval between message responses as evaluation

indicators, **a quality evaluation indicator system** for reply opinions is established, in which we determine the weight of each evaluation indicator according to the **maximum entropy model**, and finally according to the calculation The result is: 7.8536, and we can infer that the quality of the answers is good.

The algorithm involved in this article deals with the data given in the attachment more completely and removes enough contradictory data. The evaluation indicators are constructed step by step, and the processing of some small details is delicate enough, and the algorithm source code is in the calculation process are all correct.

Keywords: **Naive Bayes algorithm LDA model DBScan clustering algorithm Maximum entropy model Evaluation index system**

目录

一、问题重述

1.1 研究背景

互联网技术的飞速发展使得人们进入了大数据时代，互联网作为当今获取信息的主要渠道，与人类的关系也越来越密切。然而互联网中的绝大部分信息都是以文本形式存在，从而寻找一种能够有效处理文本数据进而对文本数据进行准确分类的方法成为当今具有重要研究价值的领域。

在社情民意了解方面，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，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

1.2 问题提出

我们需要解决的问题为以下 3 个：

- 1：群众留言分类问题

目前在处理网络问政平台的群众留言时，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。题目要求我们根据附件二中所给数据，建立关于留言内容的一级标签分类模型。另外我们需要使用 F-score 对分类的方法进行评价，判断该种分类方法是否合理，主要根据以下公式：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

其中P_i为第i类的查准率，R_i为第i类的查全率。

- 2：热点问题挖掘

某一时段内群众集中反映的某一问题可称为热点问题，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。题目要求我们根据附件 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，定义合理的热度评价指标，并给出评价结果，最后绘制出“热点问题表”以及“热点问题留言明细表”。

▶ 3：答复意见的评价

我们需要针对附件 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一套评价方案，并尝试实现。

二、问题分析

◆问题 1 分析：针对问题 1，要求建立一级标签分类模型。按照理解来说，即把留言文本信息进行归类，将留言文本信息利用算法，进行文字匹配，归到已经制定的一级标签下。我们需要对于文字匹配算法进行筛选，综合考虑其时间复杂度、准确率等因素，然后再根据 F-score 对分类方法得出的数据结果进行评价，主要考虑查准率和查全率。一般算法对于固定已有数据都可以做到查全，所以查准率显得尤为重要，这就需要我们综合考察各种算法的精度要求范围，选出合适的算法，利用合适的语言进行处理得到结果即查准率和查全率。其次，考虑 F-score 评价算法要对每一类进行求和，所以我们需要对文本数据进行预处理，考虑分类、筛选等，使得预测结果较为准确。最后得出评价结果。

◆问题 2 分析：针对问题 2，我们考虑某段时间内的热点问题。首先我们需要考虑的是以下 4 个问题：

时间要求，即这某段时间是如何选取以及具体是多少；

特定地点，即地区的选取；

留言分类，即对选取特定地点或特定人群留言问题的归类处理；

热度评价指标，即选取合理的评价指标去衡量热度；

对附件 3 所给数据我们可以选取时间长度。其次，根据留言主题我们可以筛选出每个地区的留言，然后根据问题一所构建的标签模型进行分类。最后我们需要定义合理的热度评价指标，并给出评价结果。根据附件 3 中，所给点赞数这一数据进行处理，可以选取点赞数、反对数、持续时间、出现频率等作为评价指标，再构建评价指标模型，根据评价指标模型得出热点问题表和相应热点问题对应的留言信息。

◆问题 3 分析：根据附件 4 中相关部门对留言的答复意见，对答复意见进行评价。我们需要从答复的相关性、完整性、可解释性等角度对答复意见进行评价。首先我们分析附件 4 中所给数据，答复与留言之间全是文本数据，所以考察其相关性需要根据含有特征词来判断，建立留言与答复意见之间的相关性分析，把相关性以数字直观的表示出来；其次我们需要建立以答复相关性、完整性、可解释性以及留言答复时间间隔这 4 个因素作为评价指标的评价体系，进而对答复意见的质量有一个系统性的评价。

三、符号说明

序号	符号	说明
1.	n	总留言条数
2.	N	留言类别个数
3.	R	查全率
4.	P	查准率
5.	C	留言类别集合
6.	D_i	任意一篇留言
7.	C_m	标签分类的最终结果
8.	TF_{t_k}	特征词t_k在留言D_i中出现的频数
9.	t_k	第k个特征词
10.	δ	二值函数
11.	Φ	单词在留言中的概率分布
12.	θ	概率分布矩阵
13.	W_k	权重
14.	ε	距离

四、模型的建立与求解

4.1 问题 1 模型的建立与求解

4.1.1 数据预处理

对于题目所提供的中文数据集，即附件 2 中所给留言汇总表，我们使用通用的 JIEBA 分词，得到特征词汇之后再统一去掉一些低频词语及停用词。停用词是一些没有意义的虚词和标点符号包括如‘的’，‘啊’，‘比如’，‘不但’等之类的词，停用词一般是根据领域专家构建的停用词表进行去除。在文本信息完成分词过后，需要使用向量空间模型(VSM)[1]进行文本向量化。在向量空间模型中，一篇文档对应一个高维的向量，而文档中的每个词对应向量中的一个属性。通常向量的维度为文档数据集中出现的所有词的数目，而向量每一维的取值为该单词在文档中出现的频率。接着统计该段文档中所有出现的词频，得到向量化过后的文档表示。

在分类器对文本完成分类后，需要对分类器性能进行评估。分类器的评估分为闭测试和开测试两种方式。闭测试就是将训练集的文本即使分类器的训练文本，同时还是训练集的测试文本。开放测试，则是测试集的文本与训练集的文本相互独立，没有交集。一般文本数据集较多时采用开放测试。常用的文本分类器性能评估参数有:查全率(Recall)、查准率(Precision)、FI 值、宏平均等。此外在对本题的研究中，我们采用的是查全率(Recall)和查准率(Precision)，最后利用 F-score 对所构建的模型进行评价。若有如下表 4-1 所示混淆矩阵，其中 TP 表示正确的标记为正，FP 错误的标记为正，FN 错误的标记为负，TN 正确的标记为负。

表 4-1:混淆矩阵

真实情况	**预测情况**	
	正例	反例
正例	TP	FN
反例	FP	TN

则查准率和查全率的计算公式如下：

$$P = \frac{TP}{TP + FP} \tag{4-1}$$

$$R = \frac{TP}{TP + FN} \tag{4-2}$$

其次我们对附件 2 中文本数据进行分类，根据已有一级标签分为以下 7 类：城乡建设、环境保护、交替运输、教育文体、劳动和社会保障、商贸旅游以及卫生计生。根据数据统计可得，对应总条数分别为：2009、938、613、1589、1969、877，各类标签所占比例如下图：

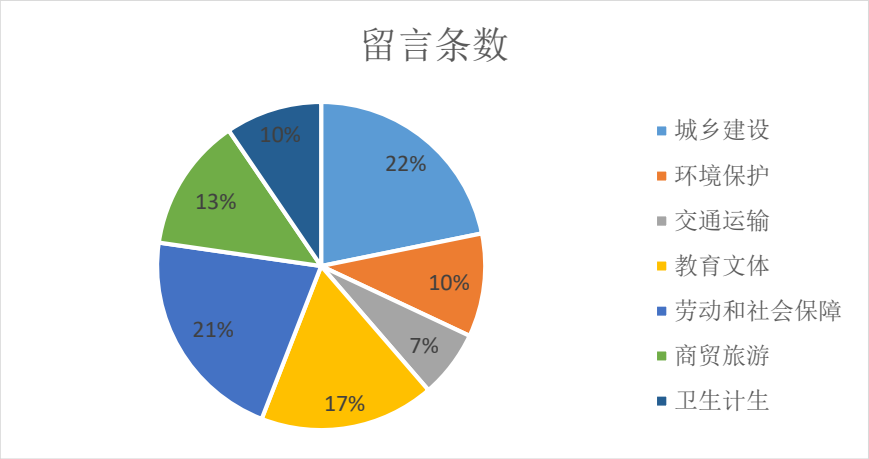

图 1：留言条数汇总图

4.1.2 多项式朴素贝叶斯模型

多项式模型[2]将文档看作一个词袋模型，认为词在一篇文档中出现的频率对文档类别的预测有影响。因此在计算条件概率的时候，多项式模型需要考虑该词出现的频率。设留言类别为 $C = \{C_1, C_2, ..., C_j\}, j = 1, 2, 3...N$，设任意一篇留言为 D_i，其中包含 m 个特征词，记为 $D_i = \{t_1, t_2..., t_m\}$，其对应的最大后验概率类别为即为留言 D_i 所属的类别，后验概率的公式为：

$$P(C_j \mid D_i) = \frac{P(D_i \mid C_j)P(C_j)}{P(D_i)} \tag{4-3}$$

其中 $P(C_j)$ 表示标签类别，$P(D_i \mid C_j)$ 表示留言 D_i 属于一级标签 C_j 的条

件概率，$P(D_i)=P(t_1,t_2,...,t_m)$表示所有特征的联合概率。贝叶斯分类的过程就是求解$P(C_j \mid D_i)$最大值的过程，对于给定的留言$P(D_i)$是一个常数，所以求解过程可以转化为下式：

$$C_m=\max_{C_j\in C} P(C_j \mid D_i)=\max_{C_j\in C} P(C_j)P(D_i \mid C_j) \tag{4-4}$$

其中C_m为标签分类的最终结果。根据朴素贝叶斯的条件独立性假设，上式可简化为

$$C_m=\max_{C_j\in C} P(C_j \mid D_i)=\max_{C_j\in C} P(C_j)P(\{t_1,t_2...,t_m\} \mid C_j) \tag{4-5}$$

$$=\max_{C_j\in C} P(C_j)\prod_{k=1}^{m} P(t_k \mid C_j) \tag{4-6}$$

普通加权朴素贝叶斯模型[3]为

$$C_m=\max_{C_j\in C} P(C_j \mid D_i)=\max_{C_j\in C} P(C_j)\prod_{k=1}^{m} P(t_k \mid C_j)^{W_k} \tag{4-7}$$

由于每次计算的概率可能比较小，为了避免出现下溢的情况，通常采取对决策规则取对数的形式，最终的判别方式为下式：

$$C_m=\max_{C_j\in C} P(C_j \mid D_i)=\max_{C_j\in C}[\ln P(C_j)+\sum_{k=1}^{m}\ln P(t_k \mid C_j)\times W_k \times TF_{l_k}] \tag{4-8}$$

式中，m表示特征词数，$t_k(k=1,2,3....m)$表示留言D_i中的第k个特征词，先验概率$P(C_j)$和条件概率$P(t_k \mid C_j)$的计算公式如下：

$$P(C_j)=\frac{\sum_{i=1}^{n}\delta(C_i,C_j)}{n+N} \tag{4-9}$$

$$P(t_k \mid C_j)=\frac{\sum_{i=1}^{n} TF_{t_k}\delta(C_i,C_j)+1}{\sum_{k=1}^{m}\sum_{i=1}^{n} TF_{t_k}\delta(C_i,C_j)+m} \tag{4-10}$$

其中n表示总的留言条数，N表示一级留言标签个数，C_i表示第i个留言的一级标签类别，TF_{t_k}表示特征词t_k在留言D_i中出现的频数，δ表示二值函数。

4.1.3 权重计算

由于朴素贝叶斯的特征条件独立性假设影响了分类器的性能，所以我们大多数使用特征加权的方式来抑制这个假设。特征词权重计算的核心思想就是对根据特征词选择算法选择出来的特征词利用某种权重计算方法赋予不同的权重，对分类贡献能力大的特征词赋以较高权重，而对分类能力较差的特征词赋予较低权重，从而让特征词具有更好的区分文本类别的能力，以达到提高分类的准确率的目的。在这里我们使用 TFIDF 来计算权重，TFIDF 权重是在文本分类中运用比较多的计算方法,TFIDF 算法的思想:特征单词在某特定文本中出现的频数越大，其对于该文本的分类作用越大，特征词在大多数文档中出现的频数越大，对于文本的分类作用越小，其结合了词频与反文档频率两者的优点。TFIDF 算法[4]将词频和反文档频率结合作为特征的权重，归一化计算方法如下：

$$IDF(t_k) = lb(\frac{n}{n(t_k)} + 0.01) \tag{4-11}$$

$$W_k = TF_{t_k} * IDF(t_k) = \frac{TF_{t_k} * IDF(t_k)}{\sqrt{\sum_{i=1}^{m}(TF_{t_k} * IDF(t_k))}} \tag{4-12}$$

其中 TF_{t_k} 表示特征词 t_k 在留言 D_i 中出现的频数，$IDF(t_k)$ 是反文档频率，n 表示总的留言条数，$n(t_k)$ 是表示出现特征词 t_k 的留言数。TFIDF 算法考虑了特征词的局部和全部分布特性。

4.1.4 实验结果及 F-score 评价

我们利用 Python 语言根据朴素贝叶斯算法进行编程，这里我们将留言内容看做：留言主题+留言详情。首先我们选择留言主题和留言详情进行分词处理，随机选取留言的一半作为训练集进行分类，其中编程如下图：

```
#导入测试集向量空间
testpath = "./testspace.dat"
test_set = readbunchobj(testpath)

##应用朴素贝叶斯算法
#alpha:0.001 alpha越小，迭代次数越多，精度越高
clf = MultinomialNB(alpha= 0.1).fit(train_set.tdm,train_set.label)

#预测分类结果
predicted = clf.predict(test_set.tdm)
total = len(predicted)
error = 0
for flabel,file_name,expct_cate in zip(test_set.label,test_set.filenames,predicted):
    if flabel != expct_cate:    #分类错误时打印
```

图 2：编程代码部分示例图

运行结果如下：

```
./test_corpus_seg/教育文体/5082.txt :实际类别:  教育文体 -->预测类别: 城乡建设
./test_corpus_seg/教育文体/5089.txt :实际类别:  教育文体 -->预测类别: 劳动和社会保障
./test_corpus_seg/教育文体/5113.txt :实际类别:  教育文体 -->预测类别: 劳动和社会保障
./test_corpus_seg/环境保护/2052.txt :实际类别:  环境保护 -->预测类别: 城乡建设
./test_corpus_seg/环境保护/2054.txt :实际类别:  环境保护 -->预测类别: 城乡建设
./test_corpus_seg/环境保护/2068.txt :实际类别:  环境保护 -->预测类别: 城乡建设
./test_corpus_seg/环境保护/2397.txt :实际类别:  环境保护 -->预测类别: 城乡建设
./test_corpus_seg/环境保护/2922.txt :实际类别:  环境保护 -->预测类别: 城乡建设
./test_corpus_seg/环境保护/2923.txt :实际类别:  环境保护 -->预测类别: 劳动和社会保障
total: 4602
error: 153
error rate: 3.3246414602346808 %
准确率: 0.9760972574248934
召回率: 0.9542785126521662
f1-score: 0.9641272903488289
```

图 3：程序运行结果图

对结果进行处理如下图：

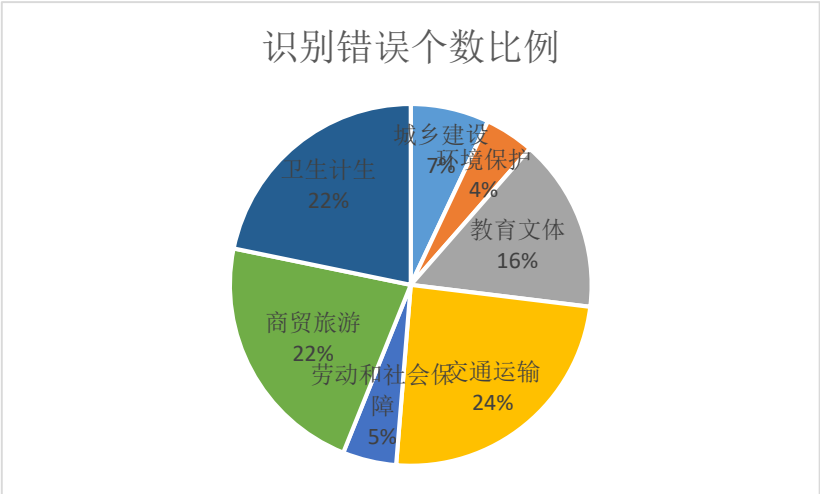

图 4：识别错误结果图

利用 F-score 评价得出如下：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

根据 Python 编程得出结果如下：

表 4-2:F-Score 结果

查全率	查准率	F-score 评价结果
0.9600558000422341	0.9782506503940348	0.9684173771741665

4.2 问题 2 模型构建与求解

4.2.1 数据预处理

由于题目要求针对特定地区或是特定人群的留言进行归类处理，这里我们选择问题 1 中建立的一级标签分类模型。对于附件 3 所给数据，我们需要筛选出特定地点或是特定人群的数据，根据留言主题以及留言详情可以对特定地区进行筛选，而对于特定人群我们可以根据问题 1 中的标签分类进行划分。首先我们根据留言主题筛选出含有明确地点的留言信息，将地点模糊、或是冲突的留言信息删除，这样便于地区的分类。其次我们利用数据筛选出 A 市和 A 市以外地区，发现 A 市留言占绝大多数，其次我们对 A 市留言进行划分，发现其可以大致分为 A1-A9 这 9 个地区。另外我们发现在伊景园属于 A 市范围，丽发新城小区属于 A2 区范围，整理汇总后，A 市分布如下：

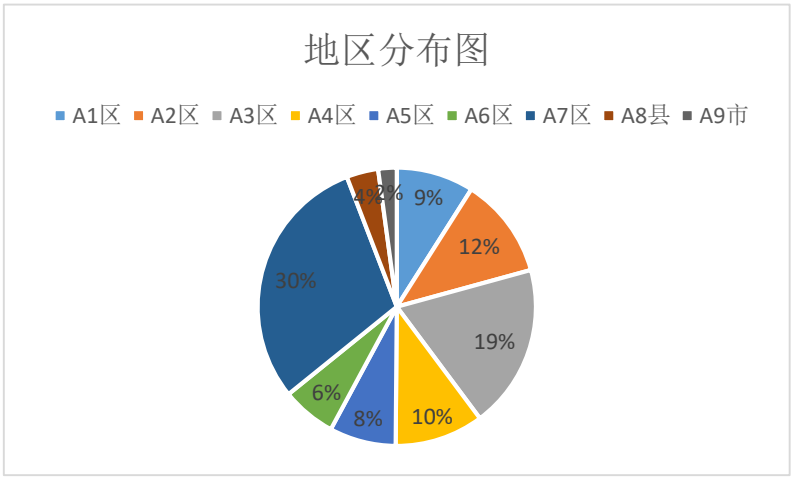

图 5：留言地区分布图

利用附件 3 所给数据，进行分类，按照问题 1 中所构建的一级标签分类模型，其分类结果如下图：

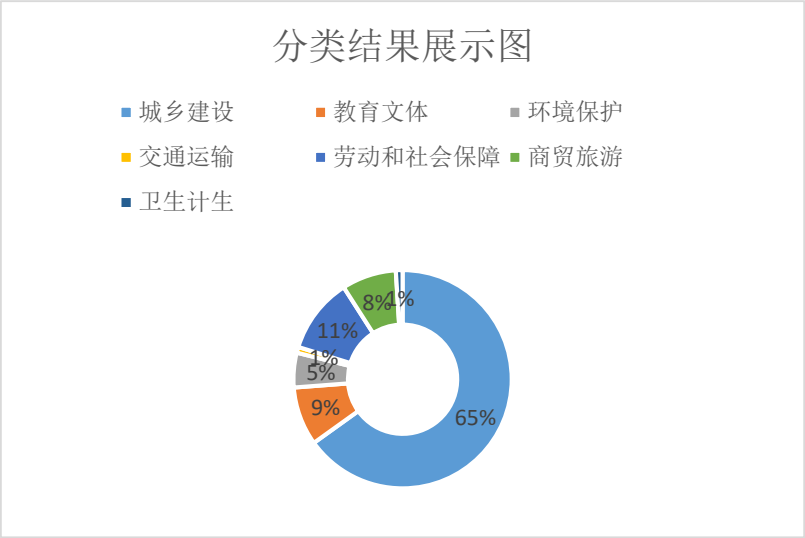

图 6：分类结果展示图

4.2.2 模型思路

针对问题 2，我们主要思路文字表述为：通过附件 2 中所给文本数据，导入生成 txt 文件，再根据问题 1 构建的一级标签分类模型，将这些文本数据分为 7 大类。然后根据 DBSCAN 聚类算法对于分成的 7 大类再进一步进行聚类处理，分成 35 小类。其次由热度值计算公式求出热度值，并进行排序。根据吉布斯采样得出 LDA 模型参数值，利用 LDA 模型抽取前五个热度留言的主题词，从而得到最终结果，详细流程图如下：

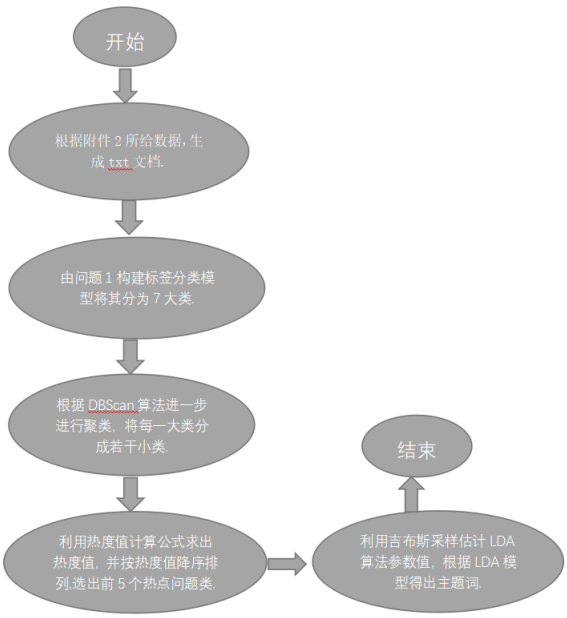

图 7：思路流程图

4.2.3 留言抽取与分类

因为附件 3 的数据没有留言类别标注，所以本文利用问题 1 建立的关于留言内容的一级标签分类模型进行分类，分类结果（部分）如下：

```
第807条留言分类为：劳动和社会保障
第808条留言分类为：商贸旅游
第809条留言分类为：城乡建设
第81条留言分类为：商贸旅游
第810条留言分类为：城乡建设
第811条留言分类为：教育文体
第812条留言分类为：城乡建设
第813条留言分类为：城乡建设
第814条留言分类为：城乡建设
第815条留言分类为：商贸旅游
第816条留言分类为：劳动和社会保障
第817条留言分类为：城乡建设
第818条留言分类为：城乡建设
第819条留言分类为：城乡建设
第82条留言分类为：卫生计生
第820条留言分类为：城乡建设
第821条留言分类为：城乡建设
```

图 8：留言分类结果（部分）

```
Python 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 03:37:03) on Windows (64 bits).
This is the Pyzo interpreter with integrated event loop for PYQT5.
Type 'help' for help, type '?' for a list of *magic* commands.
Running script: "C:\Users\pengxinyu\Desktop\数据挖掘\Code2\classify_fscore2.py"
标签分类完成
################采用分类模型对附件3分类的结果#############
分类为：交通运输：37个
分类为：劳动和社会保障：483个
分类为：卫生计生：43个
分类为：商贸旅游：349个
分类为：城乡建设：2816个
分类为：教育文体：376个
分类为：环境保护：222个
```

图 9：留言类别个数

4.2.4 新类别的进一步聚类

根据分类的结果，使每一类留言更加详细，采用 DBSCAN 文本聚类算法将类别进一步细分。

DBSCAN 算法中有两个重要参数：距离 ε 和形成簇的最小点数 minPts。

聚类从随机节点开始，保证每个节点被访问一次且仅访问一次。对每个节点，计算其范围 ε 内节点的个数。如果节点个数超过预定的点数 minPts，则该节点被标记为核心点，否则被标记为噪音点。

核心点和其范围 ε 内的点形成簇。查找簇的过程在访问所有点一遍之后完成。

算法：DBSCAN 算法描述

步骤：

While 数据库中存在未被处理的点

{从数据库中抽取一个未被处理的点

If 抽出的点周围 ε 范围内的点个数>= minPts

抽出的点 = 核心点

Else

抽出的点 = 噪音点

If 抽出的点 == 核心点

找出所有密度可达的周边的点，形成一个簇

Else

抽出的点 = 边缘点

continue

}

其核心思想为：寻找密度相连的最大集合，即从某个选定的核心对象（核心点）出发，不断向密度可达的区域扩张，从而得到一个包含核心对象和边界对象的最大化区域，区域中任意两点密度相连。

令 eps=1,min_samples=3~6，多次进行聚类，代码如图：

```
DBS_clf = DBSCAN(eps=1, min_samples=3)  #min_samples参数选取3-6多次进行聚类（手动调整）
DBS_clf.fit(weight)

#print(DBS_clf.inertia_)
print("DBSCAN分类完成")
```

图 10：DBSCAN 代码截图

4.2.5 留言评价指标建立与应用

留言的评价指标

留言的热度本质上是指某一类的留言引起的民众关注和发布数量，定量地来描述留言热度，目前学术界还没有标准的指标体系。

大多数常用的热度评估公式只考虑了话题的相对热度[6]，通过占比来体现热

度会受该时间段对应的留言总量影响，热度值不够客观，其主要原因是对于一个类别的留言而言，发布的留言数量和点赞数可以代表该话题的一个热度指标，同时，一个话题的持久性也可以代表该话题的一个热度指标。基于上述思考，本文提出了基于内容特征影响力和传播特征影响力的热度评价指标体系。

表 4-3:热度评价指标

指标	指标含义
内容特征影响力	该类留言发布数量
传播特征影响力	该类留言点赞量
	该类留言发布持续时间

从内容特征影响力来说，一个热点留言的热度代表该留言对应时间段产生。从传播特征影响力上来说，一个留言的热度代表该热度的持续时长与点赞的数量，尽管两个热点在一个月报道的数量上差不多，但是从传播特征影响力上来说数天内相关留言总量更高的热度值更高。基于上述基本思想，本文设计了基于时间消退的热度值计算方法，其公式为：

$$HotPts(T, date) = \lg(Num1_{T,date}) + \lg(Num2_{T,date}) + \alpha HotPts(T, date - 1) \quad (4\text{-}17)$$

其中 T 为任意一类指定留言；$Num1_{T,date}$为指定留言类别 T 在日期 date 下所对应的聚类簇下的文章数目；$Num2_{T,date}$为一段时间的点赞数量，α为前一段时间的留言热度值消退程度的超参数，0<α<1，一般去α=0.1；date 为系统累计统计时间，默认$HotPts(T, 0) = 0$。

采用 min-max 标准化，对所得热度值线性变换，使结果值映射到[0,1]之间转换函数如下：

$$x^* = \frac{x - min}{max - min}$$

其中 max 为样本数据的最大值，min 为样本数据的最小值。

留言分布

共轭性可以使得贝叶斯方法能够进行增量式计算，所以对于新的类别，可以使用已经训练好的 LDA 模型来预测其留言概率分布。只要认为 LDA 模型的留

言-单词概率分布是固定的，由训练语料得到的模型提供，只需求出未知类别的留言分布即可，对采样公式稍加修改便可得到对新类别的采样公式。对一种类别的采样公式如下：

$$p(z_i = k \mid w_i = t, \overrightarrow{z_{\neg i}}, \overrightarrow{w_{\neg i}}; M) = \frac{n_k^{(t)} + n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_k^{(t)} + n_{k,\neg i}^{(t)} + \beta_t)} * \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{t=1}^{V}(n_{m,\neg i}^{(k)} + \alpha_k) - 1} \quad (4\text{-}18)$$

未知类别的留言概率分布计算公式为：

$$\theta_{m_{new,k}} = \frac{n_{m_{new}}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m_{new}}^{(k)} + \alpha_k)} \quad (4\text{-}19)$$

但是使用该方法得到的新类别中不能含有训练语料库中未出现的单词。新类别留言分布的算法如下：

算法：新类别的留言分布

输入： 已经训练好的 LDA 模型，迭代次数 T。

输出：新类别的留言分布。

步骤：

1.随机初始化：对测试类别中每个单词随机赋予一个留言编号。

2.重新扫描当前类别，按照公式（5）重新采样单词的留言，并更新相关变量。

3.重复以上重新采样过程 T 次。

4.根据公式（6）计算新类别的留言概率。

本文根据附件 3 的数据，对数据集应用 LDA 模型，使用吉布斯采样[7]估计模型参数，抽取出每一类的留言及其相关文本。LDA 模型[5]相对复杂，用精确求解的方法难以求解，因此常使用近似推断的方式对 LDA 模型求解，本文使用坍塌吉布斯采样估计 LDA 模型中的参数，先将模型中的θ和$\emptyset$参数积掉，对每个单词对应的留言采样，采样收敛后通过$z_{m,n}$和$w_{m,n}$的共现关系估算θ和$\emptyset$。采样公式推导如下：

$$\begin{aligned} p(z_i = k \mid \vec{z}_{\neg i}, \vec{w}) &= \frac{p(\vec{w}, \vec{z})}{p(\vec{w}_{\neg i}, \vec{z}_{\neg i})} = \frac{p(\vec{w}, \vec{z})}{p(\vec{w}_{\neg i}, \vec{z}_{\neg i})p(w_i)} * \frac{p(\vec{z})}{p(\vec{z}_{\neg i})} \\ &\propto \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{n}_{z,\neg i} + \vec{\beta})} * \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{n}_{m,\neg i} + \vec{\alpha})} \propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_t)} * \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{t=1}^{V}(n_m^{(k)} + \alpha_k) - 1} \end{aligned} \quad (4\text{-}14)$$

其中$n_k^{(t)}$表示语料库中属于第k个留言的词项t的数量，$n_m^{(t)}$表示第m类第k个留言的数量，¬i 表示去除当前正在采样的单词。

采样收敛后通过马氏链的状态求得留言生成单词的概率和类别生成留言的概率参数集合θ和$\emptyset$，$\overrightarrow{\theta_m}$ 与$\overrightarrow{\varphi_k}$ 的估计值为

$$\varphi_{k,t} = \frac{n_k^{(t)}+\beta_t}{\sum_{t=1}^{V}(n_k^{(t)}+\beta_t)} \quad (4\text{-}15)$$

$$\theta_{m,k} = \frac{n_k^{(t)}+\beta_t}{\sum_{k=1}^{K}(n_m^{(k)}+\alpha_k)} \quad (4\text{-}16)$$

采样过程是：先对每个类别的每个单词的留言赋随机初值，然后根据吉布斯采样公式，对每个单词的留言进行采样，并更新相应参数，对一个类别中的所有单词采样完成即为一次迭代，重复迭代 1000 次，留言已经近似收敛，根据每个类别中的留言数量和每个留言下面单词的数量，估计类别生成留言的概率和留言生成单词的概率。 取若干次采样的均值，可使参数估计更加准确。采样算法如下：

算法：吉布斯采样求解 LDA 模型

输入：LDA 超参数$\vec{\alpha}$和$\vec{\beta}$，主题个数K，迭代次数T。
输出：每个类别生成留言的概率和留言生成单词的概率。
步骤：
1.随机初始化：对语料中每篇类别的每个单词，随机赋予一个留言编号；
2.重新扫描语料库，对每个单词，按照公式（1）采样它的留言，并更新相关变量；
3.重复以上重新采样过程 T 次。
4. 统计语料库中每篇类别的单词-留言矩阵，该矩阵就是 LDA 模型。
5. 根据公式（2）和（3） 计算概率参数，得到每篇类别生成留言的概率和留言生成单词的概率。

该算法的时间复杂度是O(T*N*K)，T是迭代次数，N是语料库的单词总数，K是主题个数。

留言抽取具体步骤如下：

① 利用问题 1 建立的分类模型对附件 3 中的数据进行分类；

② 使用 LDA 模型对文本语料建模，设定 LDA 模型的超参数和主题个数；

③ 使用吉布斯采样算法求解 LDA 模型的概率参数；

④ 根据求出的参数获取留言的热点词，话题在文档中出现的概率将新闻归类的相应话题中。

对于每一种类别，利用 LDA 模型无监督学习的特点，对每一类数据进行留言抽取，LDA 模型中最重要的两组参数分别是主题-词项概率分布和类别-主题概率分布，通过吉布斯采样算法可以估计参数值， 根据这两组参数可以抽取到留言热点词和与留言相关的类别。

留言热点词

LDA 模型中的参数Φ表示单词在留言中的概率分布，如图所示,同留言中的单词概率分布大不相同，且在同一留言中，概率较大的单词与该留言的意义明显相关，例如在“A3 区一米阳光婚纱艺术摄影是否合法纳税了？”留言中，出现概率最大的单词是税务局、查和纳税。在“咨询 A6 区道路命名规划初步成果公示和城乡门牌问题”留言中，概率最大的单词是成果、门牌和更换。概率较大的单词能明显地概括留言的意义，可以视作留言热点词，所以本文将每个留言的单词概率降序排序，选取留言中概率最大的 5 个特征词表示该留言。

4.2.6 结果与分析

首先多次采用 DBSCAN 模型聚类，得出如下分类结果：

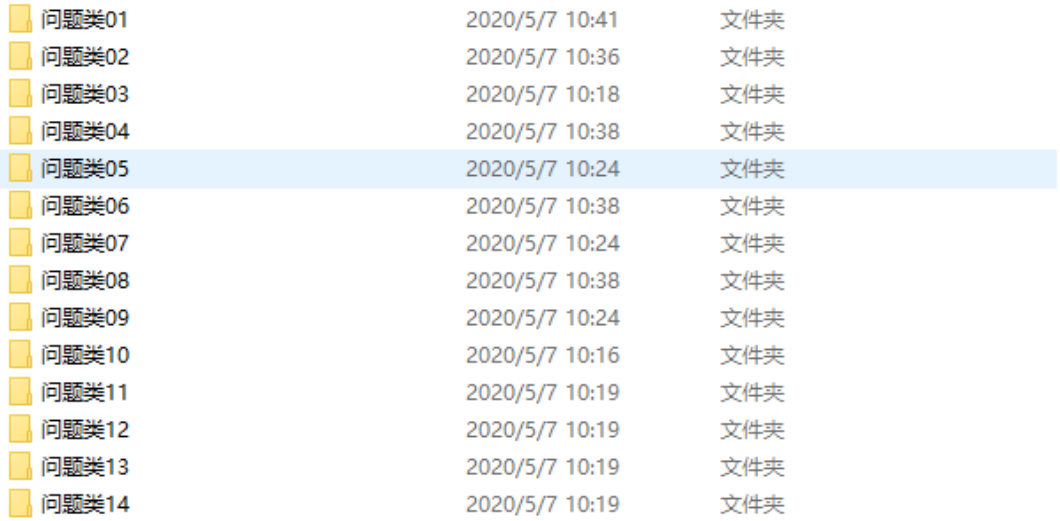

图 11：DBSCAN 模型聚类结果

根据公式（4-17），得出各个问题类的热度值如下：

图 12：各问题类的总和热度值

根据 LDA 模型，得出留言热点词分析结果如下：

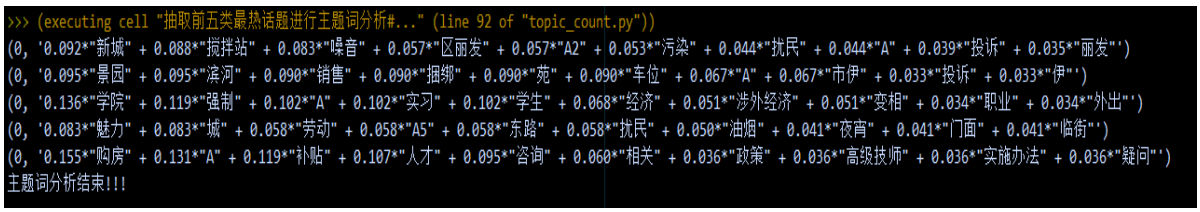

图 13：LDA 抽取留言留言热点词

按公式（4-19）计算该天的留言热度并由高到低排序，热度最高的 5 个留言如下表所示。

表 4-4:热点问题表

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	1	0.580700524298858	2019/11/2 至 2020/1/25	A 市 A2 区丽发新城	投诉搅拌站噪音灰尘污染扰民
2	2	0.4498012331122080 4	2019/7/7 至 2019/8/30	A 市滨河景园	投诉车位捆绑销售
3	3	0.4311240886812855	2017/6/8 至 2019/11/27	A 市经济学院	学院强制学生外出实习
4	4	0.3730750149656993	2019/7/21 至 2019/9/25	A 市 A5 区魅力之城	东路临街门面夜宵油烟扰民

5	5	0.2986842462890869	2019/1/16 至 2019/12/2	A 市	咨询人才购房补贴相关政策和实施办法等细节

相应热点问题对应的留言信息如下：

表 4-5：热点问题留言明细表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数
1	213464	A909233	卡丽发新城小区附近违建搅拌站噪音扰	2019/12/10	附近违建大型搅拌站。该搅拌站的设备太吵了，	0	0
1	190802	A00072636	A市丽发小区建搅拌站，噪音污染严重	2019/11/25	搅拌带来了巨大的粉尘，严重影响居民健康；同	0	0
1	214282	A909209	发新城小区附近搅拌站噪音扰民和污	2020/1/25	明，天天吵天天吵，烦死了不仅吵还臭！说好的	0	0
1	217700	A909239	丽发新城小区旁的搅拌站严重影响生活	2019/12/21	厂从绿心范围内搬迁到丽发新城小区旁边不到百	1	0
1	222831	A909228	灰尘污染的A2区丽发新城附近环保部门	2019/12/22	厂，为什么城里能修改产生大量灰尘的搅拌厂。	0	0
1	231136	A909204	诉A2区丽发新城附近建搅拌站噪音扰	2019/12/2	方建搅拌站。距离上次投诉已经过去一个月了，	0	0
1	235362	A909215	丽发新城小区附近水泥搅拌站非法经	2020/1/6	致的扬尘肆虐，严重危害居民身体健康！我们老	0	0
1	239336	A909213	市A2区丽发新城小区遭搅拌站严重污	2019/12/11	生，该厂成日运作，离居民区非常近！附近的环	0	0
1	239648	A909211	区丽发新城小区附近搅拌站明目张胆污	2020/1/6	息，还有灰尘颗粒！都不敢开窗透气了，赶紧关	0	0
1	244335	A909135	街道丽发新城社区搅拌站灰尘，噪音	2019/12/2	开发商把特大型搅拌站，水泥厂从绿心范围内搬	0	0
1	247160	A00010419	A市丽发小区建搅拌站，噪音污染严重	2019/11/25	上8点，严重影响小区居民的生活。小区业主将	0	0
1	253040	A909202	诉A2区丽发新城附近建搅拌站噪音扰	2019/12/4	老人小孩在家里根本无法正常休息！搅拌车工作	0	0
1	255008	A909208	投诉小区附近搅拌站噪音扰民	2019/11/18	是从别的地方搬过来的，体会最深的就是噪音很	0	0
1	258242	A909220	街道丽发新城社区搅拌站灰尘，噪音	2019/12/2	噪音污染严重，严重影响附近居民休息，使其不	0	0
1	259788	A909221	景云街道丽发新城社区搅拌厂危害居民	2019/12/7	灰尘，噪音污染严重，危害附近小区里的居民身	0	0
1	261072	A909207	投诉小区附近搅拌站噪音扰民	2019/11/23	部门回复，在居住区建立搅拌站水泥厂是否合法	9	2
1	264944	A0004260	丽发新城附近修建搅拌厂噪音、灰尘	2019/11/2	竟然堂而皇之修建搅拌厂，请问环保部门、城建	0	0
1	266665	A00096279	投诉小区附近搅拌站噪音扰民	2019/12/4	响了周边居民的正常生活。想问政府没有没有这	0	0
1	267050	A909227	灰尘污染的A2区丽发新城附近已扰乱居	2019/11/2	的噪音，小区居民不能正常休息，灰尘导致呼吸	0	0
1	268300	A909225	新城附近修建搅拌厂噪音污染导致生	2019/11/25	而皇之修建搅拌厂，请问是谁审批的，这不是正	0	0
1	273282	A909226	发新城附近修建搅拌厂烟尘滚滚，声	2019/12/25	音刺耳，请问政府，我们该怎么生活。审批的、	0	0
1	274004	A00026895	街道丽发新城社区附近搅拌站噪音污	2019/12/21	导反映开发商在居民区附近建搅拌站,每天噪音影	0	0
1	199379	A00092242	丽发新城附近修建搅拌厂，严重污染	2019/11/25	个别居民因此还得了疾病住院，该地区作为长	0	0
1	189950	A909204	诉A2区丽发新城附近建搅拌站噪音扰	2019/11/13	不到百米的地方建搅拌站。可想而知，一个大型	0	0
1	208285	A909205	投诉小区附近搅拌站噪音扰民	2019/12/15	休息，关闭门窗还是有很大噪音，吵得不能入睡	24	0
1	208714	A00042017	新城附近修建搅拌站，污染环境，是	2020/1/2	区内空气质量和声环境质量急剧下降，我们不是	4	0

由留言的特征词可见，留言 1 是关于丽发新城小区附近搅拌站噪音扰民的留言，留言 2 是伊景园滨河苑小区捆绑销售车位的留言，留言 3 是 A 市经济学院强制学生外出实习的留言，留言 4 是魅力之城小区一楼的夜宵摊油烟扰民的留言，留言 5 是 A 市咨询人才购房补贴相关政策和实施办法等细节的留言，相关留言均与该留言密切相关，对留言进行热度排序，热度与关于该留言的数量、时间与点赞量有关，关于该留言的数量越多、点赞量越多、时间越长，留言的热度越大。实验表明该方法具有较好的实际应用效果。

4.3 问题 3 的求解

4.3.1 聚类求解相关性

考虑留言主题、留言详情以及答复意见之间的相似程度，我们主要考虑利用聚类分析[8]来解决。在对留言以及答复进行训练后得到每个文本数据的概率分布矩阵θ；统计学上常用 JS 距离来衡量两个不同的分布，故选择它作为衡量留言和答复距离的关键指标。JS 是由 KLD 发展而来的，是基于信息熵的概念定义的，可以衡量相同时间空间里两个概率分布的差异情况。JS 的取值是对称并且有界的，当两个分布相同时，JS 取值为 0，当两个分布完全不同时，JS 取值为 1。JS 距离的计算公式如下所示：

$$JSD(\theta_p \| \theta_q) = \frac{1}{2} D(\theta_p \| \theta_m) + \frac{1}{2} D(\theta_q \| \theta_m) \quad (4\text{-}20)$$

$$(\theta_p \| \theta_q) = \sum_{x \in X} \theta_p(x) \ln \frac{\theta_p(x)}{\theta_q(x)} \quad (4\text{-}21)$$

其中 $\theta_m = \frac{1}{2}(\theta_p + \theta_q)$，$\theta_p$ 和 θ_q 分别是第 p 篇留言和第 q 篇答复的概率分布。

初始对给定的数据进行凝聚，距离越近就越容易被归为一类，直到满足一定条件为止。在文本分析中,最初将每个文档看作单独的一簇,然后依次根据距离最近的文档进行合并,直到与待合并的文档距离大于给定的阈值为止。本文通过计算JS距离度量文档之间的相似性,进而对2817条留言与答复意见进行凝聚层次聚类,并通过可视化的方式展示聚类结果。具体的算法步骤为:

①将每个留言都看成一个簇;

②计算留言答复之间的 JS 距离;

③当留言 i 文本文档和答复 j 的文本文档之间的 JS 距离小于等于阈值时,将两篇文本合并为一簇;

④重复步骤直到所有满足该条件阈值的留言合并完;

⑤增大阈值,按上述方法继续合并,直到所有的留言与答复都合并完成。在合并过程中,两簇之间的距离取值为两簇中文档 JS 距离的平均值。

通过最后结果分析留言与答复的相关性。

4.3.2 最大熵模型

根据附件 4 所给数据以及问题 3 中所知，我们选取答复的相关性、完整性、可解释性以及留言答复时间间隔作为答复意见评价的指标，并用最大熵模型[9]确定各个指标的权重。根据附件 4 中所提供数据，我们可以初步得出留言答复时间间隔平均为 20.342 天。

①答复相关性

留言与答复的相似度，两者相似度越高，则该答复越有可能是高质量答复，计算公式如式（4-20）所示。

②答复完整性

答复越完整，答复的质量越高。即答复包含的内容种类越多，所含特征词越多，

答复的质量越高。

③答复可解释性

答复对于留言的回答越合理，则答复的质量越高。

④留言答复时间间隔

时间间隔越短，说明答复越快，重视程度越高，答复质量越高。

最大熵模型的计算公式为：

$$P(d \mid s,p,r,a)=\frac{1}{R(s,p,r,a)}=e^{\lambda_1(s,d)+\lambda_2(p,d)+\lambda_3(r,d)+\lambda_4(a,d)} \quad (4\text{-}22)$$

其中归一化因子为：

$$R(s,p,r,a)=\sum\nolimits_s e^{\lambda_1(s,d)+\lambda_2(p,d)+\lambda_3(r,d)+\lambda_4(a,d)} \quad (4\text{-}23)$$

s,p,r,a 为上述的四个评价指标，d 是需要排序的答案。

通过计算得到上述四个指标的权重为：0.3567,0.2456,0.1635,0.2342。将这个四个指标标准化处理之后，利用线性加权可以得到在 10 分制基础下，该评价体系得分为：7.8536。

部分运行结果展示：

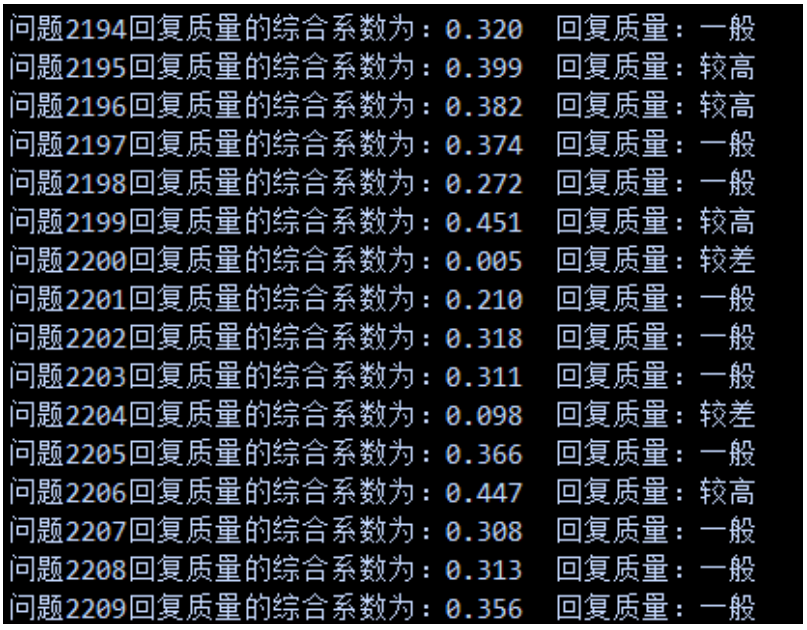

图 14：部分运行结果展示图

五、模型的评价与推广

5.1 模型的优点

①朴素贝叶斯分类模型由于其简单性、高效性和有效性被广泛由于解决文本分类问题，朴素贝叶斯文本分类问题需要考虑特征之间的独立性，属于离散型问题，由于多项式朴素贝叶斯模型在大数据集上往往表现得更好，并且总体而言多项式朴素贝叶斯模型在分类精度上要优于伯努利朴素贝叶斯模型，所以本文采用的朴素贝叶斯模型为多项式朴素贝叶斯模型。对小规模的数据表现良好，能够处理多分类任务，适合增量式训练，尤其是数据量超过内存时，可以逐批地去增量训练。

②本文深入研究了 LDA 模型以及吉布斯采样算法，LDA 模型是一种可作为特征抽取的技术，可以提高数据分析过程中的计算效率，对于不适用与正则化的模型，可以降低因维度灾难带来的过拟合。LDA 模型相对复杂，用精确求解的方法难以求解，因此常使用近似推断的方式对 LDA 模型求解，在 LDA 原始论文中使用的是变分推断，该方法推导过程复杂，算法复杂度较高，因此本文使用坍塌吉布斯采样估计 LDA 模型中的参数。

③由于分类后的数据仍然较分散，因此本文采用 DBSCAN 模型[10]对其进行二次分类，DBSCAN 对于数据库中样本的顺序不敏感，即 Pattern 的输入顺序对结果的影响不大，且以发现任意形状的簇类、能够识别出噪声点。

5.2 模型的缺点

①朴素贝叶斯模型给定输出类别的情况下，假设属性之间相互独立，这个假设在实际应用中往往是　成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好；而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有关朴素贝叶斯之类的算法通过考虑部分关联性适度改进。

②DBSCAN 模型不能很好反映高维数据和数据集变化的密度，如果样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差。

5.3 改进方向

①本文使用的 TFIDF 算法为传统的算法，忽略了特征词的类内、类间分布信

息对文本分类结果的影响，因此，可以从两个维度对特征信息增益的计方式进行改进，基于 IGDC 特征加权的朴素贝叶斯文本分类，通过全新的方式计算特征在每个类别和每个文档中的信息增益，并通过线性归的方式结合了两个维度的信息。

②针对基于密度的 DBSCAN 聚类算法，由于传统的 DBSCAN 算法在全局参数 eps 与 min_samples 选择上需人工干预以及区域查询方式过程复杂和查询易丢失对象，因此，可根据 KNN 分布与数学统计分析自适应计算出最优全局参数 eps 与 min_samples，避免聚类过程中的人工干预，实现聚类过程的全自动化。

参考文献

[1]一种改进的基于神经网络的文本分类算法[J]. 丁振国,黎靖,张卓. 计算机应用研究. 2008(06)

[2]多项式朴素贝叶斯文本分类算法改进研究[D]. 张伦干.中国地质大学 2018

[3]基于朴素贝叶斯的文本分类算法研究[D]. 何伟.南京邮电大学,2018.

[4]基于信息增益与信息熵的 TFIDF 算法[J]. 李学明,李海瑞,薛亮,何光军. 计算机工程. 2012(08)

[5]基于爬虫和 LDA 的新闻话题挖掘[D]. 曹牧原.河北大学,2018.

[6]基于数据挖掘的网络新闻热点发现系统设计与实现[D]. 童昱强.北京邮电大学,2019.

[7]基于吉布斯采样结果的主题文本网络构建方法[J]. 张志远,杨宏敬,赵越.计算机工程,2017,43(06):150-157.

[8]基于文本挖掘的网络科学会议主题研究[J]. 李小珂,赵紫娟,郭强,刘建国,李仁德.复杂系统与复杂性科学,2018,15(03):27-38.

[9]基于混合式的社区问答答案质量评价模型[J]. 袁健,刘瑜.计算机应用研究,2017,34(06):1708-1712.

[10]结合 DBSCAN 聚类算法和粒子群算法的大规模路径优化方法研究[J]. 丁乔,李旭,王建春.物流科技,2020,43(04):10-15.

附录

附录 1：

问题 1 部分代码：（朴素贝叶斯算法，运行环境为 Windows10 系统，Python3.6）

```
from sklearn.naive_bayes import MultinomialNB#导入多项式贝叶斯算法包
import pickle
from sklearn import metrics
from sklearn.metrics import precision_score #sklearn 中的精准率
#读取 Bunch 对象
def readbunchobj(path):
    file_obj = open(path,"rb")
    bunch = pickle.load(file_obj,encoding="utf-8")
    file_obj.close()
    return bunch
# 定义分类精度函数
def metrics_result(actual,predict):
    print("准确率:",metrics.precision_score(actual,predict,average='macro'))
    print("召回率:", metrics.recall_score(actual, predict,average='macro'))
    print("fl-score:", metrics.f1_score(actual, predict,average='macro'))
#导入训练集向量空间
trainpath = "./tfdifspace.dat"
train_set = readbunchobj(trainpath)
#导入测试集向量空间
testpath = "./testspace.dat"
test_set = readbunchobj(testpath)
##应用朴素贝叶斯算法
#alpha:0.001 alpha 越小，迭代次数越多，精度越高
clf = MultinomialNB(alpha= 0.1).fit(train_set.tdm,train_set.label)


#预测分类结果
```

```
predicted = clf.predict(test_set.tdm)
total = len(predicted)
error = 0
error_dict = {}
error_list = []
rlabel = ""
for flabel,file_name,expct_cate in zip(test_set.label,test_set.filenames,predicted):
    if flabel != expct_cate:       #分类错误时打印
        if flabel != rlabel:
            error_dict[flabel] = []
            error_dict[flabel].append(expct_cate)
        else:
            error_dict[flabel].append(expct_cate)
        rlabel = flabel
        error += 1
        print(file_name,":实际类别：",flabel,"-->预测类别:",expct_cate)
print("total:",total)
print("error:",error)
for label in error_dict:
    print("\n" + label + "类别 分类错误的总数为" + str(len(error_dict[label])) + "\n其中:")
    for item in error_dict[label]:
        if item not in error_list:
            error_list.append(item)
    for item in error_list:
        print("错误地将 " + label + " 分类为 "+ item + " 的个数为：" + str(error_dict[label].count(item)))
    error_list = []
print("error rate:",float(error)*100/float(total),"%")
metrics_result(test_set.label,predicted)
```

问题 2 部分代码：（DBScan 聚类算法，运行环境为 Windows10 系统，Python3.6）

```
import os
import shutil
import time
from sklearn import feature_extraction
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN
#循环进行 DBScan 聚类算法
operate_path = "./classify_result/城乡建设/6/"
for i in range(5):  #循环 5 次，获取前五类主要问题类别
    corpus = [] #建立空语料库
    file_list = os.listdir(operate_path)      #每次循环获取文件夹下的所有文件名，存入列表
    for file_path in file_list:        # 遍历类别目录下文件
        fullname = operate_path + file_path     # 拼出文件名全路径
        fp = open(fullname,'r',encoding='utf-8')
        words = fp.read()
        corpus.append(words.strip())
    print("建立语料库完成")
    ##################计算权重######################
    vectorizer=CountVectorizer()#该类会将文本中的词语转换为词频矩阵，矩阵元素 a[i][j]  表示 j 词在 i 类文本下的词频
    transformer=TfidfTransformer()#该类会统计每个词语的 tf-idf 权值
    tfidf=transformer.fit_transform(vectorizer.fit_transform(corpus))#  第 一 个 fit_transform 是计算 tf-idf，第二个 fit_transform 是将文本转为词频矩阵
    weight=tfidf.toarray()#将 tf-idf 矩阵抽取出来，元素 a[i][j]表示 j 词在 i 类文本中的 tf-idf 权重
```

```
print("产生权重完成")
DBS_clf = DBSCAN(eps=1, min_samples=3)  #min_samples 参数选取 3-6 多次进行聚类（手动调整）
DBS_clf.fit(weight)
print("DBSCAN 分类完成")
########对问题进行聚类，重新分类存放##############
value_list = list(DBS_clf.labels_)  #将聚类结果的标号列表化
question_path = "./question_dir_城乡建设/"
q_num = ''
file_list = os.listdir(operate_path)
os.makedirs(question_path + "问题类" + str(i+1) + "/")
for j in range(len(value_list)):
    if value_list[j] == 0:
        shutil.copy(operate_path + file_list[j],question_path + "问题类" + str(i+1) + "/")
        time.sleep(0.1)
        os.remove(operate_path + file_list[j])
```

问题 3 部分程序：评价体系算法

```
import os
import jibe
import re
from  gensim import corpora,models,similarities #语料库方法,模型方法,相似度方法
from collections import defaultdict #计算单词频率
#############判断问题与回复文本的相似度(答复的相关性)###############
question_path = './question_seg/'
answer_path = './answer_seg/'
i = 1
simi_max = 0
doc_max = 0
```

```
num_max = 0
link_max = 0
words_max= 0
for filename in os.listdir(question_path):
    quname = question_path + filename
    fp = open(quname, 'r', encoding= 'utf-8')
    content = fp.readlines()
    fp.close()
    qu_tests = [jieba.lcut(text.strip()) for text in content]
    L1 = len(qu_tests[0])
    L2 = len(qu_tests[1])
    dictionary=corpora.Dictionary(qu_tests)
    anname = answer_path + filename
    fp = open(anname, 'r', encoding= 'utf-8')
    content = fp.read()
    fp.close()
    an_text_word = jieba.lcut(content.strip())
    doc_vectors=[dictionary.doc2bow(text) for text in qu_tests]
    tfidf=models.TfidfModel(doc_vectors)
    tfidf_vectors = tfidf[doc_vectors]
    new_vec=dictionary.doc2bow(an_text_word)
    feature_Num=len(dictionary.token2id.keys())

index=similarities.SparseMatrixSimilarity(tfidf[doc_vectors],num_features=feature_N
um)   #稀疏矩阵相似度
    sim=index[tfidf[new_vec]]   #通过 tfidf 和要对比的文本的稀疏向量计算相似
度
    #print("回复相似度:" ,(sim[0]*L1 + sim[1]*L2)/(L1+L2))
    #回复可解释性，统计文本中的条文个数，时间日期，链接数量
    simi = (sim[0]*L1 + sim[1]*L2)/(L1+L2)
```

```
        doc = re.findall(re.compile(r'[《].*?[》]', re.S),content)
        num = re.findall('\d+',content)
        link = re.findall(re.compile(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+'),content) #  匹配模式
        if simi>simi_max:
            simi_max = simi
        if len(doc) > doc_max:
            doc_max = len(doc)
        if len(num) >num_max:
            num_max = len(num)
        if len(link) >link_max:
            link_max = len(link)
        #将文本分词后的长度作为回复完整性的依据
        if len(an_text_word) > words_max:
            words_max = len(an_text_word)
target_list = []        #回复质量指标列表
for filename in os.listdir(question_path):
        quname = question_path + filename
        fp = open(quname, 'r', encoding= 'utf-8')
        content = fp.readlines()
        fp.close()

        qu_tests = [jieba.lcut(text.strip()) for text in content]
        L1 = len(qu_tests[0])
        L2 = len(qu_tests[1])
        dictionary=corpora.Dictionary(qu_tests)

        anname = answer_path + filename
        fp = open(anname, 'r', encoding= 'utf-8')
```

```
        content = fp.read()
        fp.close()

        an_text_word = jieba.lcut(content.strip())
        doc_vectors=[dictionary.doc2bow(text) for text in qu_tests]
        tfidf=models.TfidfModel(doc_vectors)
        tfidf_vectors = tfidf[doc_vectors]

        new_vec=dictionary.doc2bow(an_text_word)

        feature_Num=len(dictionary.token2id.keys())

index=similarities.SparseMatrixSimilarity(tfidf[doc_vectors],num_features=feature_Num)  #稀疏矩阵相似度
        sim=index[tfidf[new_vec]]  #通过 tfidf 和要对比的文本的稀疏向量计算相似度
        #回复可解释性，统计文本中的条文个数，时间日期，
        simi = (sim[0]*L1 + sim[1]*L2)/(L1+L2)    #相似度
        #统计条文数量，时间数字，链接个数
        doc = re.findall(re.compile(r'[《].*?[》]', re.S),content)
        num = re.findall('\d+',content)
        link = re.findall(re.compile(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+'),content) #  匹配模式

        #综合回复指标,为灰度相似度，回复可解释性、回复完整性分配权重，计算综合指数
        #0-max 标准化（0-Max Normalization）
        target = (simi/simi_max)*0.5 + ((len(doc)/doc_max)*0.4+(len(num)/num_max)*0.4+(len(link)/link_max)*0.2)*0.3 +
```

```
(len(an_text_word)/words_max)*0.2
        target_list.append(target)
        #print(target)
#############
#打印回复质量
target_min = min(target_list)
target_max = max(target_list)
index = 1
for target in target_list:
        Prop = (target-target_min)/(target_max-target_min)   #根据回复的综合系数在最大综合系数差中所占的比例，为回复质量定等级
        if Prop >0.6:
                print("问题"+str(index)+"回复质量的综合系数为："+str(target)[0:5] + "回复质量：较高")
        if 0.3<Prop <=0.6:
                print("问题"+str(index)+"回复质量的综合系数为："+str(target)[0:5] + "回复质量：一般")
        if Prop <= 0.3:
                print("问题"+str(index)+"回复质量的综合系数为："+str(target)[0:5] + "回复质量：较差")
        index += 1
```

附录 2：热点问题留言明细表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数
1	213464	A909233	投诉丽发新城小区附近违建搅拌站噪音扰民	2019/12/10 12:34:21	近违建大型搅拌站。该搅拌站的设备太吵了，	0	0
1	190802	A00072636	A市丽发小区建搅拌站，噪音污染严重	2019/11/25 18:58:05	拌带来了巨大的粉尘，严重影响居民健康；同	0	0
1	214282	A909209	A市丽发新城小区附近搅拌站噪音扰民和污染环境	2020/01/25 09:07:21	了，天天吵天天吵，烦死了不仅吵还臭！说好的	0	0
1	217700	A909239	丽发新城小区旁的搅拌站严重影响生活	2019/12/21 02:33:21	从绿心范围内搬迁到丽发新城小区旁边不到百	1	0
1	222831	A909228	噪音、灰尘污染的A2区丽发新城附近环保部门不作为	2019/12/22 10:23:11	，为什么城里能修改产生大量灰尘的搅拌厂。	0	0
1	231136	A909204	投诉A2区丽发新城附近建搅拌站噪音扰民	2019/12/02 11:20:21	方建搅拌站。距离上次投诉已经过去一个月了，	0	0
1	235362	A909215	暮云街道丽发新城小区附近水泥搅拌站非法经营何时休	2020/01/06 20:45:34	效的扬尘肆虐，严重危害居民身体健康！我们	0	0
1	239336	A909213	A市A2区丽发新城小区遭搅拌站严重污染	2019/12/11 11:44:11	，该厂成日运作，离居民区非常近！附近的环	0	0
1	239648	A909211	A市A2区丽发新城小区附近搅拌站明目张胆污染环境	2020/01/06 22:41:31	是，还有灰尘颗粒！都不敢开窗透气了，赶紧关	0	0
1	244335	A909135	A市暮云街道丽发新城社区搅拌站灰尘，噪音污染严重	2019/12/02 12:11:23	开发商把特大型搅拌站，水泥厂从绿心范围内搬	0	0
1	247160	A00010419	A市丽发小区建搅拌站，噪音污染严重	2019/11/25 19:08:41	上8点，严重影响小区居民的生活。小区业主将	0	0
1	253040	A909202	投诉A2区丽发新城附近建搅拌站噪音扰民	2019/12/04 12:10:21	老人小孩在家里根本无法正常休息！搅拌车工	0	0
1	255008	A909208	投诉小区附近搅拌站噪音扰民	2019/11/18 12:23:22	是从别的地方搬过来的，体会最深的就是噪音很	0	0
1	258242	A909220	A市暮云街道丽发新城社区搅拌站灰尘，噪音污染严重	2019/12/02 12:23:11	噪音污染严重，严重影响附近居民休息，使其不	0	0
1	259788	A909221	A市暮云街道丽发新城社区搅拌厂危害居民健康	2019/12/07 00:00:00	尘，噪音污染严重，危害附近小区里的居民身	0	0
1	261072	A909207	投诉小区附近搅拌站噪音扰民	2019/11/23 23:12:22	部门回复，在居住区建立搅拌站水泥厂是否合	9	2
1	264944	A0004260	A2区丽发新城附近修建搅拌厂噪音、灰尘污染	2019/11/02 14:23:11	然堂而皇之修建搅拌厂，请问环保部门、城	0	0
1	266665	A00096279	投诉小区附近搅拌站噪音扰民	2019/12/04 17:23:22	响了周边居民的正常生活。想问政府没有没有	0	0
1	267050	A909227	噪音、灰尘污染的A2区丽发新城附近已扰乱居民生活	2019/11/02 10:18:00	的噪音，小区居民不能正常休息，灰尘导致呼	0	0
1	268300	A909225	A2区丽发新城附近修建搅拌厂噪音污染导致生活不正常	2019/11/25 10:17:58	皇之修建搅拌厂，请问是谁审批的，这不是正	0	0
1	273282	A909226	A2区丽发新城附近修建搅拌厂烟尘滚滚，声音刺耳	2019/12/25 10:17:59	音刺耳，请问政府，我们该怎么生活。审批的	0	0
1	274004	A00026895	A市暮云街道丽发新城社区附近搅拌站噪音污染严重	2019/12/21 10:11:09	争反映开发商在居民区附近建搅拌站,每天噪音	0	0
1	199379	A00092242	A2区丽发新城附近修建搅拌厂，严重污染环境	2019/11/25 10:17:56	个别居民因此还得了疾病住院，该地区作为	0	0
1	189950	A909204	投诉A2区丽发新城附近建搅拌站噪音扰民	2019/11/13 11:20:21	不到百米的地方建搅拌站。可想而知，一个大	0	0
1	208285	A909205	投诉小区附近搅拌站噪音扰民	2019/12/15 12:32:11	休息，关闭门窗还是有很大噪音，吵得不能入睡	24	0
1	208714	A00042015	A2区丽发新城附近修建搅拌站，污染环境，影响生活	2020/01/02 00:00:00	区内空气质量和声环境质量急剧下降，我们不	4	0
2	218709	A00010669	A市伊景园滨河苑捆绑销售车位	2019/08/01 22:42:21	购房者签订正规购房合同，强制收取18万5千元	1	0
2	218739	A909184	A市伊景园 滨河苑欺诈消费者	2019/08/24 00:00:00	，诱骗购房者交付车位定金，还不写入协议，	0	0
2	223247	A00044759	投诉A市伊景园滨河苑捆绑销售车位	2019/07/23 17:06:03	希望能取消车位捆绑。2.希望这个房子对职工来	0	0
2	224767	A909176	伊景园滨河苑车位捆绑销售！广铁集团做个人吧！	2019/07/30 14:20:08	不给我合同，说什么预购不用！后面就没有消	0	0
2	230554	A909174	投诉A市伊景园滨河苑捆绑车位销售	2019/08/19 10:22:44	位费用是伤筋动骨的，购房者中有一部分人是	0	0
2	234633	A909194	无视消费者权益的A市伊景园滨河苑车位捆绑销售行为	2019/08/20 12:34:20	商要求职工购买房子的同时一对一购买所谓按	0	0
2	236301	A909197	和谐社会背景下的A市伊景园滨河苑车位捆绑销售	2019/08/30 16:32:12	的车位，这个还是和谐社会么？政府的正经职	0	0
2	244243	A909198	关于伊景园滨河苑捆绑销售车位的投诉	2019/08/24 18:23:12	地属于业主共有，但属于城镇公共绿地和明示	0	0
2	251601	A909187	A市伊景园滨河苑诈骗钱财	2019/08/01 22:42:21	者18万5千元的认购金后不与购房者签订合同，	0	0
2	255507	A909195	违反自由买卖的A市伊景园滨河苑车位捆绑销售行为	2019/08/20 12:34:21	时一对一购买所谓按成本价销售的12万一个的	0	0
2	195511	A909237	车位捆绑违规销售	2019/08/16 14:20:26	位我们反映多次一直没有人彻查处理，到底是	0	0
2	258037	A909190	投诉伊景园滨河苑捆绑销售车位问题	2019/08/23 11:46:03	现在楼盘还未建成,广铁集团却要求捆绑购买12	0	0
2	258386	A909185	A市伊景园滨河苑欺压百姓	2019/08/28 00:00:00	购买其根本不需要的车位，不买就取消购房资	0	0
2	196264	A00095080	投诉A市伊景园滨河苑捆绑车位销售	2019/08/07 19:52:14	资格，国家三令五申禁止捆绑车位销售，为何	0	0
2	268299	A909193	惊！！A市伊景园滨河苑商品房竟然捆绑销售车位	2019/08/21 15:32:33	协商要求职工购买房子的同时一对一购买所谓	0	0
2	271517	A909238	开发商联合广铁集团捆绑车位销售	2019/08/11 12:02:27	强制要求铁路职工买房必须购买车位要不然不	0	0
2	276460	A909170	A市伊景园滨河苑捆绑销售车位是否合理?	2019/08/24 17:23:11	都是没有车的并没有购买车位的需求。网上查	0	0
2	289950	A00044759	投诉A市伊景园滨河院捆绑销售车位	2019/07/07 07:28:06	房资格的铁路职工一对一购买车位的协议几个	0	0
2	205277	A909234	伊景园滨河苑捆绑车位销售合法吗？！	2019/08/14 09:28:31	苑楼盘时捆绑购买12万一个的车位，不买车位	1	0
2	205982	A909168	坚决反对伊景园滨河苑强制捆绑销售车位	2019/08/03 10:03:10	格，这是明显的违法捆绑销售！通过购买车位	2	0
2	207243	A909175	伊景园滨河苑强行捆绑车位销售给业主	2019/08/23 12:16:03	强迫购买车位，不买车位就取消购买资格，这	0	0
3	215507	A00010323	A市五矿万境K9景存在严重的消防安全隐患	2019/09/12 14:48:07	宣传的完全不一致。长廊没有任何防护措施。质	1	0
3	208069	A00094436	A5区五矿万境K9县的开发商与施工方建房存在质量问题	2019/05/05 13:52:50	地位？！这种房屋质量是否能符合五矿在西地省	2	0
3	208636	A00077171	A市A5区汇金路五矿万境K9县存在一系列问题	2019/08/19 11:34:04	区也曾发生过狗咬人，请问有人对养宠物的情	2097	0
4	224042	A00014225	咨询A市人才购房及购房补贴实施办法等相关问题	2019/01/16 11:58:48	居住证。因为换了工作单位，2018年08月27日	0	0
4	225657	A00051791	关于A市人才购房补贴的疑问	2019/02/25 14:43:15	政策鼓励其在A市安家发展，而对于先买房后	6	2
4	265577	A0004787	咨询A市人才购房补贴通知问题	2019/12/02 11:57:49	户其在A市城内工作，年龄35岁以下全日制博	1	0
4	270015	A00020115	为何我的A市人才购房补贴申请不通过?	2019/05/29 16:51:26	到已经结婚，属于夫妻共同财产，也没有注	0	0
4	280288	A00041301	咨询A市购房政策的社保缴纳相关问题	2019/08/01 15:05:54	。本人今年毕业已经在A市找到工作并入职，现	0	0
4	282104	A00090921	关于高级技师申报A市人才新政购房补贴的相关问题咨询	2019/07/29 10:42:38	填职工社会保险参保材料；是否要求社保必须	1	0
4	282248	A00010609	咨询A市人才购房补贴事宜	2019/01/30 15:42:19	在后才可以？个人觉得，公布的政策上并没有	0	0
4	283494	A00085185	《A市人才购房及购房补贴实施办法（试行）》相关问题咨	2019/07/30 19:06:40	受户籍和个税、社保缴存限制。本人有技师证	0	0
4	203760	A00011195	咨询A市人才购房补贴政策	2019/06/25 15:43:23	名义购买了社保，社保信息齐全，资料齐全，	0	0
4	204245	A00024579	关于A市高级技师购房补贴的疑问	2019/03/01 10:58:43	窗口电话0000-00000000，得到的答复仍然是	14	0
5	195095	A00039089	魅力之城小区临街门面油烟直排扰民	2019/09/05 12:29:01	熏死人。一天24小时都是烟。请政府关闭处理	3	0
5	269977	A00042313	A3区梅溪湖看云路一师润芳园小区临街门面油烟直排扰民	2019/09/05 12:29:01	无证经营，长期油烟烧烤熏死人。一天24小时都	0	0
5	284147	A909113	区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空	2019/07/21 10:29:36	还是觉得要维护社会和谐稳定，合法维权。为	3	0
5	360100	A324156	魅力之城小区临街门面油烟直排扰民	2019/09/05 12:29:01	熏死人。一天24小时都是烟。请政府关闭处理	0	3
5	360101	A324156	A5区劳动东路魅力之城小区油烟扰民	2019/07/28 12:49:18	烟机清洗也没有。每天油烟直排。熏死树木。	0	4
5	360102	A1234140	A5区劳动东路魅力之城小区底层餐馆油烟扰民	2019/09/10 06:13:27	油烟扑进屋内，窗户长期不能打开，晚上营业到	0	0
5	360103	A0012425	A5区劳动东路魅力之城小区临街门面烧烤夜宵摊	2019/09/25 00:31:33	作为烧烤夜宵更加扰民，油烟24小时量死人	0	1
5	360107	A0283523	区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空	2019/07/21 10:29:36	还是觉得要维护社会和谐稳定，合法维权。为	0	3
5	360108	A0283523	东路魅力之城小区一楼的夜宵摊严重污染附近的空气，急	2019/08/01 16:20:02	还是觉得要维护社会和谐稳定，合法维权。为	0	6
5	360109	A0080252	万科魅力之城小区底层门店深夜经营，各种噪音扰民	2019/09/04 21:00:18	邻充斥着吆喝声、拼酒声、炒菜烧烤的锅铲炭火	0	0