“智慧政务”中的文本挖掘应用

摘要

本文旨在基于某政务平台所给出的大量群众民意、群众留言的文本数据，结合文本内潜在的关键词信息，利用自然语言处理技术，将文本数据化，通过特征工程、数据分析与挖掘，对留言分类、热点问题探索和答复评价这三个问题提出针对性的解决方案，对提高政府的管理水平和提升施政效率起推动作用。

针对问题一，对群众留言按照给定的一级标签分类成各个类别，以便后续分派至相应的职能部门。对文本进行预处理后，将文本转化为数据，构建出词频矩阵，基于 TF-IDF 方法改变特征的权重，最终将文本向量化。基于样本所显示出的各种特性，选用随机森林模型进行分类，随后通过卡方过滤样本特征和调整模型的参数对模型提优，得到最终的模型，并输出混淆矩阵，得出 F1-score。

针对问题二，挖掘热点问题，使各部门能有针对性的处理问题，提高服务效率。通过设计一套基于问题的留言次数、点赞数和反对票的打分系统来评判问题是否为热点问题，同时，结合利用 DBSCAN 聚类法和 TruncatedSVD 降维技术，将相似的问题归为一类，视为同一种问题，以便于计算问题的留言次数。最后通过打分排名给出最终的热点问题。

针对问题三，对答复意见的质量给出一套评价方案。考虑使用余弦相似度将留言与答复作对比，根据两者的相似性做出评分。以及考虑根据答复时间长短对答复的及时性做一个评分。

关键词：TF-IDF；随机森林；卡方过滤；DBSCAN 聚类；TruncatedSVD 降维；

Abstract

Based on the text data of mass public opinion and mass message given by a certain government platform, combining with the potential keyword information in the text, using the natural language processing technology, the text will be digitized, through feature engineering, data analysis and mining, to the message classification, hot issue exploration and response evaluation of the three issues targeted solutions, to improve the management level of the government and improve the efficiency of governance.

In response to question 1, the mass messages are classified into categories according to the given level 1 label for subsequent assignment to the corresponding functional department. After preprocessing the text, the text is transformed into data, the word frequency matrix is constructed, and the weight of the feature is changed based on the TF-IDF method, and the text is finally vectorized. Based on the characteristics of the samples, the random forest model is selected for classification, and then the model is optimized by chi-square filtering and adjusting the parameters of the model, the final model is obtained, and the confusion matrix is output, and the F1-score. is obtained.

Aiming at the problem two, we can excavate the hot issues so that all departments can deal with the problems pertinently and improve the service efficiency. To judge whether the problem is a hot issue by designing a scoring system based on the number of messages, the number of likes and the number of votes against the problem. At the same time, combining the DBSCAN clustering method and the TruncatedSVD dimension reduction technology, the similar problems are classified as the same problem, so as to facilitate the calculation of the number of messages on the problem. Finally, the final hot issues are given by scoring ranking.

In response to question 3, a set of evaluation programmes are given for the quality of the responses. Consider using cosine similarity to compare messages with responses, and score according to their similarity. and consider a rating of the timeliness of the response based on the length of the response.

Key words: TF-IDF; random forest; chi-square filtering; DBSCAN clustering; TruncatedSVD dimensionality reduction;

目录

一、问题分析

近年来，随着网络问政平台逐步成为政府了解民意的重要渠道，并且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。利用自然语言处理和文本挖掘技术可以减轻工作量，并且提高工作效率，有助于施政水平的提高和民意了解的深入。

问题总共给出四个附件。附件一为标签的集合，该标签用于后续分类各个群众的留言，并为留言打上分类标签。附件二包括群众的网络 id、留言主题、留言时间、留言详情和留言分类，较为重要的属性为留言主题、留言详情和留言分类。附件三包括的信息和附件二相同，区别在于附件三没有分类标签，取而代之的是点赞数和反对数，用于作为后续的热点问题挖掘的中间指标。附件四是群众留言及时间和答复及时间的一张表。

总体来看，问题一的留言分类和问题二的热点问题挖掘，都需要将文本数据化，只有将文本数据化以后，才能利用模型或算法，而自然语言处理技术将文本数据化是一个难点，所以问题一二的难点实际上是对文本数据的处理。

针对问题一，对群众留言进行一级标签分类，这很明显是一个多分类的问题，目前有许多分类模型，我们要根据数据特性合理选择模型。主要步骤基本如下：对数据进行去重操作、将文本数据化、选择合适的模型、把数据带入模型得出结果、合理调整参数提升模型优度。

针对问题二，在 4000 多个留言文本中挖掘出热点问题，并按照热点顺序排名，同时将地名/人群识别出来。何为热点问题？为此我们事先自定义一个热度指标，该指标是基于留言次数，点赞数和反对票的，那么，留言次数的计算就是本问题的重点。也就是说，我们要在 4000 个留言中寻找出相似的文本，实际上它们是同一个问题，只是由于留言群众的不同而表达上有不同。主要的方法是利用无监督学习聚类算法，将聚在同一类的问题视为相似的留言。

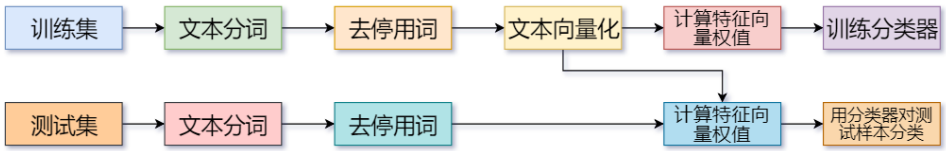
针对问题三，根据相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出一套评价方案。分析基于所给数据的留言与答复意见的文字未能发现显著相似性，部分答复内容为将留言问题转交给其他部门或暂无解决方案。考虑到该答复为人工答复，所属部门或所属职位确实能力有限，所处情况确实短时间内无法给出真正解决问题的答复，但从该部门本身的

情况来看确实尽心尽力。而除去该类留言后对剩余留言和答复使用余弦相似度计算文本相似度，结果偏差较大，不能作为预想指标。最终只考虑计算答复及时性作为指标进行评分。

二、解决问题

2.1 留言分类

该问题的流程大概如图所示，接下来按照流程一步一步解决问题。



2.1.1 前期准备

1）数据去重

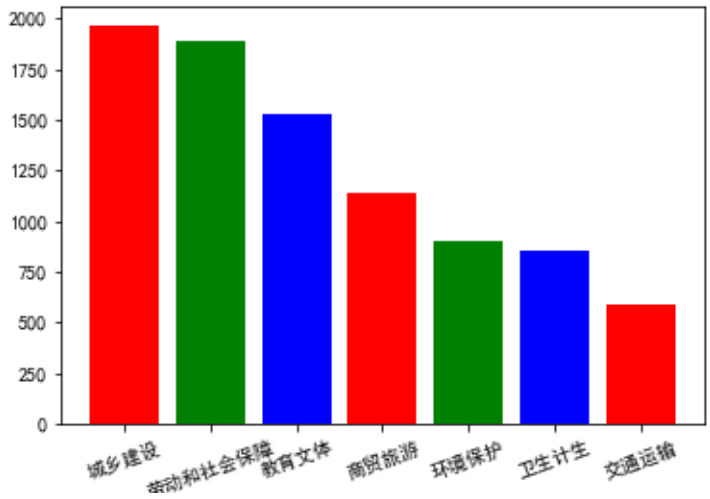
经过对附件二样本的初步探查，发现有些重复的数据，在留言主题和留言详情中都有体现。第一个步骤就是删除这些重复数据，对于有监督的分类模型，去除掉重复的数据可以减少样本数量，提高模型计算效率，并且可以防止过拟合，提高模型的泛化能力。通过该操作，附件二的数据量最终为 8866。

华庭 小区 高层 二次 供水 楼顶 水箱 长年 不洗 自来水 龙头 水 霉味 日常生活 必不...
华庭 小区 高层 二次 供水 楼顶 水箱 长年 不洗 自来水 龙头 水 霉味 日常生活 必不...

2）查看类别分布

我们根据各个类别的数据在去重之后的分布情况，有利于我们观察哪些类别偏多，哪些类别偏少，或是类别之间差距不大、比较均衡。由于在分类模型中，有些模型对类别不均衡敏感，而有些不敏感，掌握分布情况有利于我们选择模型，或是对样本进行欠采样、过采样等操作使得我们可以使用对类别敏感的模型。通

过绘制柱形图可以观察分布情况，如图所示：



可以看到类别是不均衡的。

3）编码标签

通过查看类别分布发现样本中总共有 7 类，通过编码函数 LabelEncoder() 将七个类别分别将文字转化成 0、1、2、3、4、5、6，以便于后续的模型使用。

4）去除无效字符

在“留言详情”这一列中，有许多除了汉字以外的字符，例如英文、数字，这些字符对文本的实质内容没有什么太大的作用，也就是说这些字符都不会成为该文本的关键词。把这些字符去掉，可以降低后续文本数据化的数据维度，提升模型效率。

5）分词

文本是连续的词语组成的句子，它们都是由不同的词性依次组合成句子，例如“我在吃饭”可以分成“我”（人称代词）、“在”（介词）、“吃饭”（动词），通过将一个完整的句子划分成一个一个的词语，目的是为了后续将文本向量化做准备，一个句子难以量化，而当这个句子划分成一个一个词语组成的时候，这些词语就相当于向量里的元素，所以分词成为了最不可或缺的基本步骤。

利用 python 的 jieba 分词库，可以将文本划分为一个一个的词语，以留言主题为例，展示部分分词结果：

留言主题
[A, 市, 西湖, 建筑, 集团, 占, 道, 施工, 有, 安全隐患]
[A, 市, 在水一方, 大厦, 人为, 烂尾, 多年, , , 安全隐患, 严重]
[投诉, A, 市, A1, 区苑, 物业, 违规, 收, 停车费]
[A1, 区, 蔡锷, 南路, A2, 区华庭, 楼顶, 水箱, 长年, 不洗]
[投诉, A, 市, 盛世, 耀凯, 小区, 物业, 无故, 停水]
[咨询, A, 市, 楼盘, 集中, 供暖, 一事]
[A3, 区, 桐梓, 坡, 西路, 可可, 小城, 长期, 停水, 得不到, 解决]
[反映, C4, 市, 收取, 城市, 垃圾处理, 费, 不, 平等, 的, 问题]
[A3, 区, 魏家坡, 小区, 脏乱差]
[A2, 区, 泰华, 一村, 小区, 第四届, 非法, 业委会, 涉嫌, 侵占, 小区业主...
[A3, 区梅, 溪湖, 壹号, 御湾, 业主, 用水, 难]
[A4, 区鸿涛, 翡翠, 湾, 强行, 对, 入住, 的, 业主, 关水, 限电]
[地铁, 5, 号线, 施工, 导致, A, 市锦楚, 国际, 星城, 小区, 三期, 一个...
[A6, 区润, 和, 紫, 郡, 用电, 的, 问题, 能, 不能, 解决]
[A, 市锦楚, 国际, 新城, 从, 6, 月份, 开始, 停电, 好, 多次, 了]
[给, A9, 市, 城区, 南, 西, 片区, 城铁, 站, 设立, 的, 建议]

6）去除停用词

在将句子分词之后，我们就可以把句子向量化了，但是在向量化之前，我们还可以优化分词结构，把例如“的”、“了”、“和”这一类的停用词给删除，因为这些停用词和之前去除无效字符的作用相同，他们对于一个句子的实质内容是没有贡献的，少了他们句子的主要意思不会改变，删除它们可以进一步的减少向量的维度，避免维度过高，不利于模型的使用。中文停用词在互联网上又很多开源的资源，利用这些资源，保存在一个 csv 文件中，就可以对文本进行去除停用词的操作，部分结果展示如图：

市 西湖 建筑 集团 占 道 施工 安全隐患
市 在水一方 大厦 人为 烂尾 多年 安全隐患

投诉 市 盛世 耀凯 小区 物业 无故 停水
咨询 市 楼盘 供暖 一事

市 中南大学 楚雅 医院 强制 心梗 患者 体内 安放 支架 漠视 病人 健康

建议 市 地铁 号 线西 延 二期 暂缓 修建 改建 中速 磁悬浮
建议 修建 市 城铁 站 市 火车 南站 市 黄花 国际 机场 高铁 线

7）划分训练集与测试集

在后续的建模中，使用训练集的数据来训练模型，然后用测试集上的误差作为最终模型在应对现实场景中的泛化误差。这里选取的比例为 80%训练集和 20%的测试集，测试集在原数据中随机抽样，剩下的 80%为训练集。

2.1.2 文本向量化

1）构建词频矩阵

对训练样本中的词汇，计算它在该训练集文本中出现的频率，从而将每个样本的留言详情向量化，各个样本向量组成词频矩阵。利用 Sklearn 中的 CountVectorizer 提取文本特征，对于每一个训练文本，它只考虑每种词汇在该训练文本中出现的频率。CountVectorizer 会将文本中的词语转换为词频矩阵，它通过 fit_transform 函数计算各个词语出现的次数。

对测试样本中的词汇，采取同样的操作，利用 CountVectorizer 提取文本特征，构成词频矩阵。由于测试集只占 20%的样本，特征数必然小于训练集，使得两个数据集特征维度不一样，不利于后续测试集带入模型测试。所以在给测试样本构建词频矩阵的时候，需要利用 vocabulary 参数，让测试集的维度与训练集保持一致。

2）改变特征权重

选用 TF-IDF 方法改变特征权重。

TF-IDF（词频-逆文档频率）是一种统计方法，用以评估一个单词在一个文档集合或语料库中的重要程度。TF 代表词频，IDF 代表逆文档频率，其主要的思想可以解释为：如果某个词或短语在一篇文章中出现的频率高（即 TF 高），并且在其他文章中很少出现（即 IDF 高），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。

实际上的做法是$TF * IDF$，其中：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = log\frac{|D|}{|\{j: t_i \epsilon d_j\}|}$$

$n_{i,j}$ 是词语 t_i 在文档 d_j 中的出现次数，分母是文档 d_j 中所有词语的出现次数之和

$|D|$ 是语料库中所有文档总数，分母是包含词语 t_i的所有文档数

通过 Sklearn 中的 TfidfTransformer 类，统计 CountVectorizer 中每个词语的 TF-IDF 权值，最后获得最终的向量，该向量即为文本的数据化表示。该向量最终将用于带入模型。

2.1.3 构建模型

1）选择模型

目前有许多机器学习的分类模型，它们各自有各自的优缺点，选择合适的模型解决问题是关键的一步。文本分类问题上常用的机器学习分类模型有朴素贝叶斯、支持向量机、KNN 算法和随机森林等，其中，朴素贝叶斯已被广泛应用于垃圾邮件、垃圾短信的识别和分类上，可见他在文本分类应用上的强大能力。

根据附件二的数据情况，属于大样本数据，在去重后仍然有 8866 个数据，并且留言详情较长，将留言详情向量化后向量特征较多、维度较高，另外，还存在着类别分布不均衡的问题。根据数据的这些特性，我们可以去选择合适的模型进行分类。

首先，KNN 算法对样本分类不均衡敏感，并且计算量较大，不太适合这一问题的环境。而支持向量机虽然在解决高维问题上有很好的表现，但在大样本环境下的能力不如其他的模型。

对于随机森林而言，是一个性能强大的分类模型，在当前的很多数据集上，相对于其他算法都有着很大的优势，表现良好。它的许多特性都符合该问题的数据特性。如：

① 随机森林能够处理很高维度（特征很多）的数据，并且无需特征选择。

② 对不平衡的数据不敏感。

③ 训练速度快，避免样本过多模型效率低。

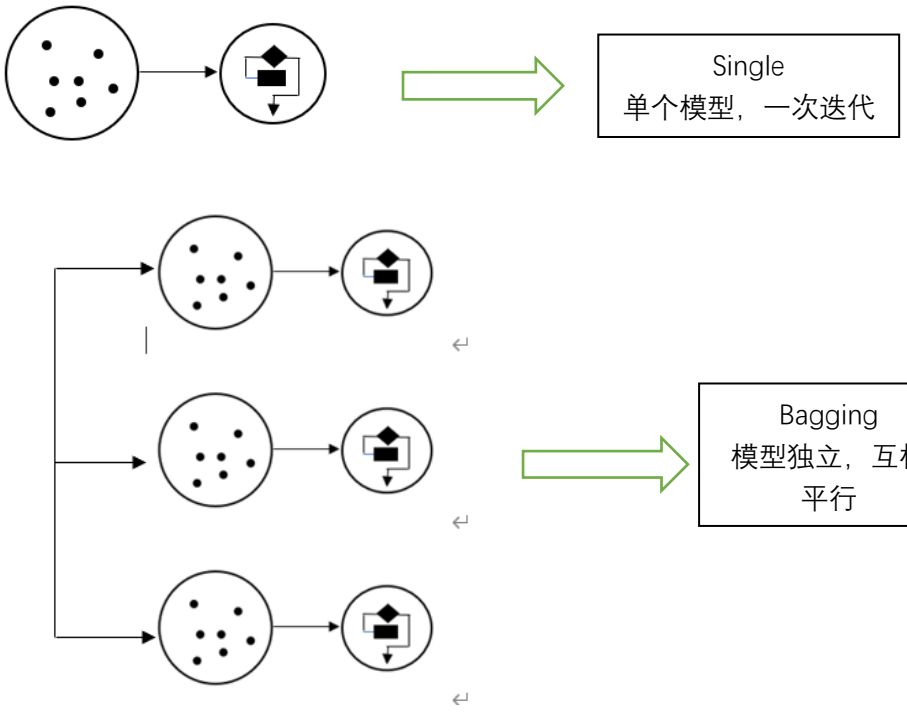
④ 抗过拟合能力强。

所以，针对于问题一，随机森林模型是一个较好的选择。

2）工作原理

随机森林的算法基于决策树，是决策树的改进，是一种袋装集成学习（bagging）的思想。决策树往往会产生过拟合问题，尤其会发生在存在整组数据的决策树上。随机森林是多重决策树的组合，不只是一棵决策树，而是构建多

个相互独立的决策树，多数表决原则来决定集成评估器的结果。随机森林算法下决策树的数量越多，泛化的结果更好。



基本原理可以解释为：

① 从数据集中随机选择 k 个特征，共 m 个特征（k≤m）。根据这 k 个特征建立决策树。

② 重复 n 次，这 k 个特性经过不同随机组合建立起来 n 棵决策树。

③ 对每个决策树都预测结果。存储所有预测的结果，从 n 棵决策树中得到 n 种结果。

④ 将得到高票数的预测目标作为随机森林算法的最终预测。

3）建立模型

通过Sklearn库中的RandomForestClassifier类，初步建立随机森林模型，先不调整参数，使用默认的参数，将之前已经处理好的向量带入模型即可。

2.1.4 结果分析

1）初次结果

通过将之前处理好的文本向量带入随机森林模型中，树设定为默认的 100，设置随机模式 90，同时采用袋外数据和测试集数据分别查看模型的得分情况，

袋外数据的准确度有 84.487%、测试集上的准确度有 85.016%。

```
# 建立随机森林分类器
rfc = RFC(n_estimators=100,random_state=90,oob_score=True).fit(x_vec,y)
rfc.oob_score_ , rfc.score(x_test_vec , y_test)    #袋外数据得分与测试集得分
```

```
(0.8448697068403909, 0.8501628664495114)
```

可以看到，袋外数据的准确度和测试集上的准确度相差不大。由随机森林的特性，一棵树的生成过程并不会使用所有的样本，通过未使用的样本（袋外样本）可以评估这个树的准确度，其他子树叶按这个原理评估，最后取平均值。因为袋外样本的存在，因此不需要进行交叉测试，该测试集上的得分可以看成较为稳定、正确的得分。

2）输出混淆矩阵并计算各类指标

混淆矩阵，即误差矩阵，是一种分类模型的评判指标。混淆矩阵的每一列代表了预测类别，每一列的总数表示预测为该类别的数据的数目；每一行代表了数据的真实归属类别，每一行的数据总数表示该类别的数据实例的数目，每一列中的数值表示真实数据被预测为该类的数目。

以二分类问题为例，混淆矩阵可以表示为：

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP Type I
	Negative	FN Type II	TN

- 真实值是 positive，模型认为是 positive 的数量（True Positive=TP）
- 真实值是 positive，模型认为是 negative 的数量（False Negative=FN）这就是统计学上的第二类错误（Type II Error）
- 真实值是 negative，模型认为是 positive 的数量（False Positive=FP）这就是统计学上的第一类错误（Type I Error）
- 真实值是 negative，模型认为是 negative 的数量（True Negative=TN）

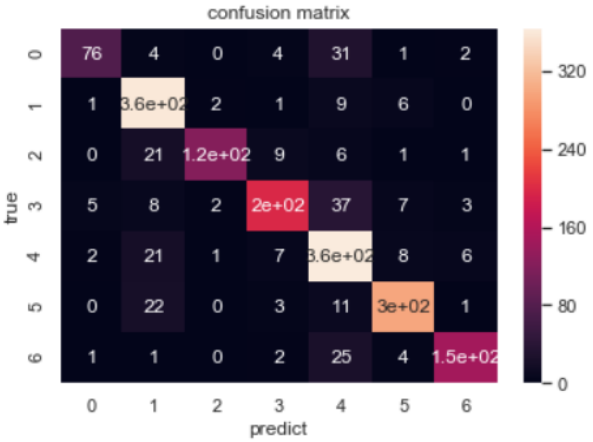
有了混淆矩阵，可以算出各类指标：

	公式	意义
精确率 PPV	$\frac{TP}{TP+FP}$	在模型预测是 Positive 的结果中，模型预测对的比重
召回率 TPR	$\frac{TP}{TP+FN}$	在真实值是 Positive 的结果中，模型预测对的比重
F1-Score	$\frac{2*PPV*TPR}{PPV+TRP}$	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。

对于多分类问题而言，$F1-Score=\frac{1}{n}\sum_{i=1}^{n}\frac{2*PPV*TPR}{PPV+TRP}$

从上一步的初步带入模型得出的结果，我们可以输出混淆矩阵及混淆矩阵的热力图：

```
[ 76,   4,   0,   4,  31,   1,   2]
[  1, 361,   2,   1,   9,   6,   0]
[  0,  21, 121,   9,   6,   1,   1]
[  5,   8,   2, 198,  37,   7,   3]
[  2,  21,   1,   7, 364,   8,   6]
[  0,  22,   0,   3,  11, 299,   1]
[  1,   1,   0,   2,  25,   4, 147]
```



再通过 Sklearn 中的 classification_repor 的函数可以输出基于混淆矩阵的报告，报告展示如图：

	precision	recall	f1-score	support
class 0	0.89	0.64	0.75	118
class 1	0.82	0.95	0.88	380
class 2	0.96	0.76	0.85	159
class 3	0.88	0.76	0.82	260
class 4	0.75	0.89	0.82	409
class 5	0.92	0.89	0.90	336
class 6	0.92	0.82	0.86	180
micro avg	0.85	0.85	0.85	1842
macro avg	0.88	0.82	0.84	1842
weighted avg	0.86	0.85	0.85	1842

其中画红色方框的数值即为该问题初步使用模型的 F1-Score 得分值，可以看到 0.84 这一比较高的分数，说明模型表现良好。

3）卡方过滤

初步带入模型时的特征维度非常高，虽然随机森林在处理高维数据时有优秀的表现，但是向量维度过高，可能携带了一些不太重要的特征，影响模型的学习，此外，向量维度过高也会降低模型的计算效率，这在往后的调参操作耗费大量的时间。采用卡方过滤法来选择特征，降低特征空间维度。

卡方过滤是专门针对分类问题的。卡方检验类计算每个非负特征和标签之间的卡方统计量，并依照卡方统计量由高到低为特征排名，借此除去最可能独立于标签，与我们分类目标无关的特征，其原假设为特征与标签独立，借助于 P 值，我们将过滤掉 P 值大于显著水平 0.05 的特征，降低特征空间的维度。过滤后训练集的特征空间的维度从 68097 下降到了 786，过滤效果非常显著。

```
# 卡方过滤筛选p小于0.05的特征
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

chivalue, pvalues_chi = chi2(x_vec, y)
k = (pvalues_chi<=0.05).sum()
x_fchi = SelectKBest(chi2 , k).fit_transform(x_vec , y)
x_fchi.shape
```

```
(7368, 786)
```

将过滤后的特征重新带入之前建立的随机森林模型中训练，同时将测试集以训练集的 P 值同时过滤特征，重新查看模型得分。再次运行模型的时候出分结果非常快，显著提高了模型的效率。

```
# 利用过滤后的训练数据建立随机森林
rfc = RFC(n_estimators=100,random_state=90,oob_score=True).fit(x_fchi,y)
rfc.oob_score_   #查看袋外数据得分
```

```
0.8429695982627579
```

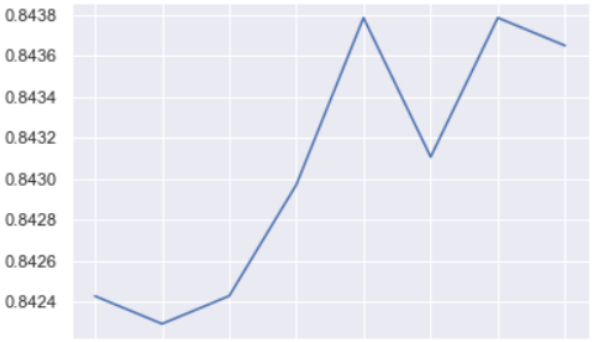
```
# 取出测试集对应部分的特征，对测试集进行预测
x_test = x_test_vec[:,(pvalues_chi<=0.05)]
rfc.score(x_test , y_test)
```

```
0.8501628664495114
```

从得分结果来看，和过滤之前的得分几乎没有变化，该过滤方法至少没有降低模型的得分，还显著提高了模型的效率。

4）模型调参

随机森林的主要参数是 n_estimators、max_features、min_sample_leaf 和 max_depth,上一步卡方过滤后模型的训练速度加快了很多,所以可以利用循环不断调整这些参数，寻求最佳的组合，同样是利用袋外数据和测试集数据得分作为指标。



（循环调整 n——estimators 可视化图）

最终得出了 n_estimators=100、max_features=11，其他参数采用默认设置的组合，该组合下模型在袋外数据准确度为 85.328%，测数集数据准确度为 86.808%，比未调参之前稍有提高。再查看一下混淆矩阵的报告，可以看出，F1-Score 也有轻微的提高。

	precision	recall	f1-score	support
class 0	0.92	0.68	0.78	133
class 1	0.90	0.95	0.93	401
class 2	0.90	0.79	0.84	172
class 3	0.83	0.83	0.83	243
class 4	0.78	0.87	0.82	405
class 5	0.91	0.92	0.92	302
class 6	0.93	0.86	0.89	186
micro avg	0.87	0.87	0.87	1842
macro avg	0.88	0.84	0.86	1842
weighted avg	0.87	0.87	0.87	1842

5）总结

初次尝试随机森林模型时使用初始默认参数，第一次模型运行结果为袋外数据准确度 84.487%、测试集准确度 85.016%。此次运行模型速度非常慢，约四分钟，得出的 F1-Score 为 0.84。随后采用卡方过滤大幅度降低特征空间维度，再次带入模型运行，运行速度明显提高，模型的效率得以提升，并且准确度并没有降低。最后进行模型的调参工作，最终的模型参数为 n_estimators=100、

max_features=11，其他参数采用默认设置，此时模型的结果为袋外数据准确度85.328%，测数集数据准确度86.808%，F1-Score为0.86，均有提升。

2.2 热点问题挖掘

2.2.1 前期准备

附件三的数据形式和附件二相同，我们要做的预处理和上一问题基本相同，首先,对数据去重,考虑到做热点问题挖掘时,一个用户可能会留言不同的问题，但两个留言详情完全相同的情况，几乎不可能是两个人所为，所以只根据留言详情去重即可。

接下来和问题一的步骤相同,将留言详情分词、去除停用词、构建词频矩阵、利用TF-IDF方法改变特征权重，即可完成留言详情从文本到向量的转化。

2.2.2 自定义热度指标

在观察附件三的数据后，我们认为热度指标应该由留言次数、点赞数、反对票三要素组成。

为了便于问题的统计以及排行，对题目中某些条件进行合理假设。

1）因不清楚数据是否可靠，假设点赞的网民与留言的网民之间不存在任何联系，即不存在刷票现象。

2）因为个人习惯差异，假设网民留言前已经知道其他留言里是否存在相似的问题。

3）因无法具体确认留言的网民是否会更换账号对同一问题进行不同的描述，假设和留言详情都不相同的情况下视为不同网民进行的留言。

根据点赞数的多少排序后观察发现，点赞数最多的留言有2000个点赞，而超过20个点赞的留言只有46条。考虑到在同一问题上，假如看到相似问题，人们更倾向于点赞。并且我们认为如果某一类问题只有一条留言，而它的点赞数是最高的，那么这条留言理应属于热门问题；同样的，如果其他类型的问题只有两三条，而某类问题的数量达到了几十条甚至更多，那么这类问题也理应属于热门问题。考虑到这些方面并且经过适当调整，我们最终决定热度指标如下规定：

按照一类问题留言一次加一百分，点赞一次加一分，反对一次扣一分的规则

对所有类进行排行，并且最终选取出得分最高的 5 类留言作为热门问题。

即：$P(\text{分数}) = \text{一类问题中的留言次数} * 100 + \text{点赞数} - \text{反对票}$

2.2.3 聚类分析

1）选择算法

根据上述自定义的热度指标，我们需要统计出相似留言的留言次数。那么，怎么样的留言才会是相似的留言呢？我们已经把留言详情全部转化为了向量，并且利用 TF-IDF 分配了权重，在特征空间上，两个相似的留言它们的夹角会小，也可以说他们的欧氏距离会小。但是两两计算夹角或是距离会把问题复杂化。考虑使用聚类方法，将有相同特性的向量的聚在一起，同一类的向量可以看成是同一类的问题，那么一个类里面向量的个数就是这一类问题的留言次数。

聚类是一种无监督学习的机器学习技术，它涉及到数据点的分组。给定一组数据点，我们可以使用聚类算法将每个数据点划分到不同的类。理论上，同一类中的数据点应该具有相似的属性或特征，而不同类中的数据点应该具有明显不同的属性或特征。

本问题选用 DBSCAN 聚类算法来完成对相似问题的归类。选用该算法基于以下的考虑：

① 由于我们不知道会有多少类相似的问题，试先并不能指定类的数量，如果后续一步一步调试会花费大量时间，而 DBSCAN 聚类算法不需要事先知道形成的类的数量。

② DBSCAN 算法可以形成任意形状的类。

③ 这 4000 多的数据粗略地来看，不存在大量的相似的问题，也就是说该样本噪声点数量多，DBSCAN 算法在聚类地同时还能识别出噪声点，受噪声点影响小。

④ DBSCAN 对样本地输入顺序不敏感，不同的输入顺序对结果的影响都不大，虽然作为无监督学习，但重复调试形成的结果都较为稳定。

DBSCAN 是基于密度的聚类算法，它使用一组关于“邻域”的参数来描述样本分布的紧密程度，其工作原理可分为以下步骤：

① 给定邻域距离 e 和邻域最小样本个数 MinPts。

② 遍历样本，找出所有满足邻域距离 e 的核心对象的集合。

③ 任意选择一个核心对象，找出其所有密度可达的样本并生成聚类簇。

④ 从剩余的核心对象中移除②中找到的密度可达的样本。

⑤ 从更新后的核心对象集合执行②-③步直到核心对象都被遍历或移除。

2）TruncatedSVD 降维

聚类算法在太高的维度的表现都很差，因为在高维度的空间中，所有的样本都会表现得距离倾向于彼此接近，因此数据点之间的距离关系会弱化，而该问题下留言详情的向量化特征非常多，因此在使用 DBSCAN 聚类之前，先要降低特征的维度。

TruncatedSVD 压缩原有数据的维度，寻找 r 新变量，使它们反映事物的主要特征，将特征向量的维数降低，挑选出最少的维数来概括最重要特征。

利用 Sklearn 中的 TruncatedSVD，完成对数据的降维，这个维度需要试先设定，会影响聚类的效果，可通过后续的调整来提升聚类的效果。

3）初步尝试聚类

经过多次尝试，特征维度降维至 10，邻域距离设置 0.2，邻域最小样本个数设置为 4 的效果最佳。将处理好的文本向量带入 DBSCAN 模型，查看输出结果：

```
labels = db.labels_
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
print('聚类数：',n_clusters_)

print('噪点数：',n_noise_)
```

```
聚类数：  23
噪点数：  3972
```

随机挑选几个类来查看数据：

```
dbscandf[dbscandf['cluster'] ==8]['留言主题']
```

```
231                西地省科技职业学院未经沟通就强制学生搬离宿舍
268                    A6区二中强制学生暑假补课且收费
303              A市涉外经济学院组织学生外出打工合理吗？
497                A3区实验中学组织学生有偿高价补习
700                    A6区第二中学有偿补课
709                A6区丁字中学周末补课收费
735          投诉A市电子科技职业学院要求学生办各种银行卡
902              A市长郡A1区实验中学组织学生补课
983    西地省财政经济学院食堂只开放十余个窗口，有些学生吃不上饭
1109               A市星创教育无证办学
1111               反映K1区学校食堂的一些问题
1117           A7县三中补课收费，家长负担不起
1142           A6区金海中学寒假要求补课两次
1195          西地省高考学生少数民加分名单何时公布？
1221       A市医学院的学费价格为何会比其他学校高出五六倍？
```

发现内容看起来是相似的，但是地名参差不齐，实际上是不同的问题，聚类效果不太理想。

4）改进聚类方法

通过上述尝试，我们发现有些问题内容上是相似的，但是地名上不同，这实际上是不同的问题。为解决这一问题，我们可以先根据地名，将地区划分，在不同的地区分别聚类。根据对数据的观察，附件三所给出的数据是有地点上的规律的，地名分布为：A 市、A1 区、A2 区、A3 区、A4 区、A5 区、A6 区、A7 县、A8 县和其他地区。通过 Pandas 库的操作可以将这些地区分好，分别保存为不同的 DataFrame，以 A4 区为例，重新展示聚类效果：

```
labels = db.labels_
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
print('聚类数：',n_clusters_)

print('噪点数：',n_noise_)
```

```
聚类数：  8
噪点数：  247
```

```
dbscandf[dbscandf['cluster'] ==1]['留言主题']
```

```
206                  说说A4区凯乐国际城小区周边安全隐患问题
295       A4区凯乐国际城太平路路灯失修大半年未解决，该向谁问责
595              请解决A4区凯乐国际城周边太平路的路灯问题
1854             请解决A4区凯乐国际城周边道路乱停车问题
2261          A4区凯乐国际城周边车辆乱停乱放问题依然没解决
3525                   A4区凯乐国际城周边交通乱象
3701       A4区凯乐国际城周边的庆和里路、太平路路灯很久没亮了
Name: 留言主题, dtype: object
```

可以看到，聚类效果非常好，一个类里的问题基本相似了。

将不同地区的向量依次聚类，将相似的问题按照原来附件所给的格式保存在一个 excel 表中。

2.2.4 热点问题

由上述得到的各个类别的问题，按照之前自定义热度指标进行评分计算，以 A 市统计得分为例展示，该地区类别一共有 7 类，按照自定义热度值表得分最高的为标签 2 的类，得分 1298。

```
cluster
0     400
1     500
2    1298
3     401
4     502
5     529
6     500
Name: score, dtype: int64
```

最后根据排行分数得到了热点问题表以及热点问题留言明细表，热点问题表展示如下，明细表以 excel 表保存将以附件形式上传，不作展示。

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	3559	2019/01/14	A市	58车贷案
2	2	2207	2019/07/11	伊景园滨	捆绑销售
3	3	2197	2019/8/19	A市A5区	一系列问
4	4	1857	2019/4/11	A市金毛	配套入学
5	5	1538	2019/01/06	丽发新城	搅拌站噪

热点问题表

2.3 答复意见评价

经过文本相似度对比后发现相似度并不能与相关性挂钩，最终我们只考虑及时性作为评分标准。

用 python 导入数据之后将留言时间和答复时间转化为 datetime，查看最快的回复以及最久的回复时间。

```
In  [37]: data['time'].max()
Out[37]: Timedelta('1160 days 10:28:14')

In  [38]: data['time'].min()
Out[38]: Timedelta('0 days 00:30:53')
```

将最快回复设为 1，最慢回复设为 0，对所有留言进行评分，最后部分留言评分如下图。

	留言编号	留言用户	留言主题	留言时间	留言详情	答复意见	答复时间	timegrade
1091	25582	UU008353	请求在A市椰梨街道如八字槽门一带增加一个地铁站点	2016-02-19 08:43:00	\n\t\t\t\t\t\n\t\t\t\t\t 椰梨街道 作为A7县五个街道之一 作为A...	来信人： 您好！您的来信已收到，您的诉求所涉及到的职权范围不属于A7县人民政府职权范围内，...	2016-02-19 09:13:53	1.000000
2560	133087	UU008781	L3县谭坪村廖家组人盼望有条回家路	2016-05-10 16:26:45	\n\t\t\t\t\t\n\t\t\t\t\t 我是L3县七甲坪镇谭坪村廖家组人,两年前...	“UU008781” 您好！帖文已收悉，我们将按工作程序转交至县交通局、七甲坪镇政府参考。...	2016-05-10 17:14:18	0.999990
1590	47887	UU0081332	咨询幼儿园学费的标准	2016-09-08 14:33:57	\n\t\t\t\t\t\n\t\t\t\t\t 咨询幼儿园学费的标准?幼儿园的保险费是否...	网友： 你好，详情请咨询C市教育局基教科：0000-00000000。谢谢您对C市教育的...	2016-09-08 15:30:35	0.999985
2531	131394	UU0082243	L9县县明耻巷路灯几月未亮，老百姓盼望维修	2017-11-22 16:05:10	\n\t\t\t\t\t\n\t\t\t\t\t 从L9县县西街卖炭路口进来明耻巷，紫气东...	“UU0082243” 您反映的问题已转交县住建局核实、办理。 2017年11月22日	2017-11-22 17:04:46	0.999983
1194	32712	UU0081124	B6县皇图岭镇界联村请求整改道路	2019-11-20 16:43:58	\n\t\t\t\t\t\n\t\t\t\t\t尊敬的市委书记，我是生于B6县皇图岭镇界联村...	“UU0081124”您好！您通过平台《问政西地省》的留言收悉，已转有关部门调查核处，如有相...	2019-11-20 17:48:59	0.999980