目 录

基于智慧政务的文本挖掘

随着互联网的不断发展，政府治理和公共服务的方式也发生了一定的变革，网络平台逐渐成为政府了解民意、民生的重要渠道。通过各种网络平台可以合法获取更多关于群众的问题反映的数据，对这些数据进行划分、整理，从中提取关键信息，可以帮助政府更科学、准确地了解民意和解决民生问题。

原始数据的预处理（1）**文本分词**。将数据导入 python，运用 **jieba 分词**对文本数据进行分词。(2)**去除停用词**。将出现频率多却又无实际意义的词去掉，“的”。（3）**去除特殊字符**。通过正则化方法将标点符号去掉，例如“，”、“！”，提高处理文本的效率。

针对问题一：（1）一级标签分类模型的建立。建立关于留言内容的**一级标签分类模型**，运用 **KNN 分类模型**。（2）分类模型评价方法。运用 **F-Score** 对建立的 KNN 分类模型进行评分。

针对问题二：根据留言内容使用**命名实体识别方法**识别出一段时间内的特定地点、人群反映的问题，进而对各问题进行归类、定义热点问题建立评价指标，最终筛选出群众反映的热点问题。

针对问题三：对于答复意见的质量评价方案，分别运用**词袋模型**、**TF-IDF 模型**、**LSI 模型**对答复意见文本和留言详情进行对比，检验其相似度，比对各模型的差异，最后通过**相似度百分比**对答复意见的质量进行评价。

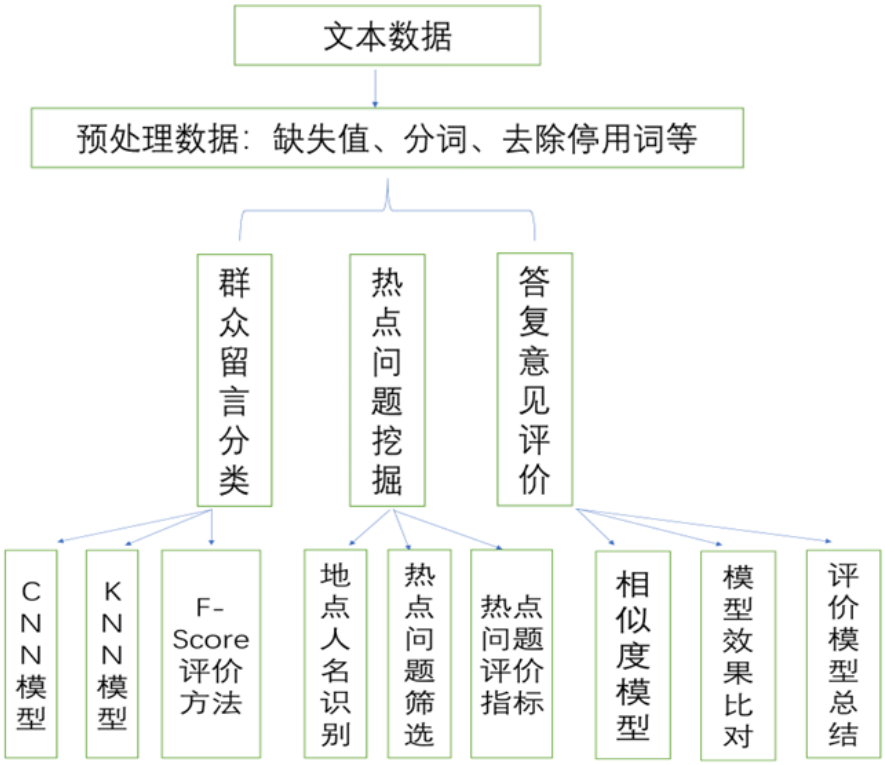
关键词： jieba 分词；KNN 分类模型；F-Score；词袋模型；TF-IDF 模型；LSI 模型

1 引言

1.1 背景介绍

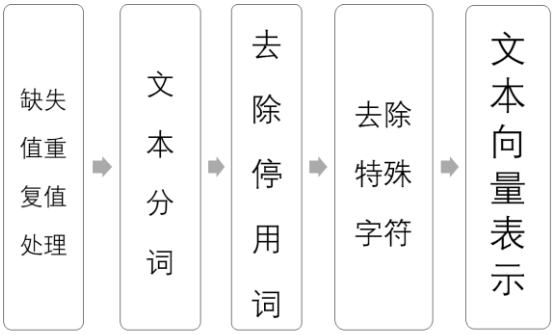
从上世纪九十年代起,我国电子政务经历了从传统政务到移动政务再到当下智慧城市大背景下催生的新型电子政务——智慧政务。近年来，我国各地政府均在建设和推行智慧政务，智慧政务是在信息化时代背景下，通过信息化手段为公众提供高效服务的政务模式，综合运用互联网和信息网络技术，以大数据为核心，推动政府廉洁高效运转、政府政策制定更加精准、服务大众更为便捷、信息化透明化程度更高等方面发挥了积极的作用。但同时在此模式运行下也产生了一些新问题，本文将基于对存在的一些问题进行探究。

1.2 目标实现流程图



2 数据预处理

本文对数据进行了预处理操作，包括对缺失值、重复值进行查看和处理；根据模型构建的要求对文本数据进行分词、去停用词以及去除特殊符号的处理，数据预处理流程图如下：



2.1 缺失值、重复值处理

导入数据并对文本数据进行检验。部分结果如图 1-1 所示：

```
In [2]: #预处理
        #检验是否有缺失值
        from pandas import DataFrame
        from pandas import Series
        data1.isnull().sum()
        #检验是否有重复值
        from pandas import DataFrame
        from pandas import Series
        data1.duplicated().sum()
Out[2]: 0

In [3]: data2.isnull().sum()
        data2.duplicated().sum()
Out[3]: 0

In [4]: data3.isnull().sum()
        data3.duplicated().sum()
Out[4]: 0

In [5]: data4.isnull().sum()
        data4.duplicated().sum()
Out[5]: 0
```

图 2-1

从结果中看到，该文本数据中无缺失值与重复值。

2.2 文本分词

本文运用 jieba 分词进行中文分词。jieba 分词主要是基于统计词典，构造一个前缀词典；然后利用前缀词典对输入句子进行切分，得到所有的切分可能，

根据切分位置，构造一个有向无环图；通过动态规划算法，计算得到最大概率路径，也就得到了最终的切分形式。

Jieba 分词支持三种分词模式：

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。

文本分词根据具体问题可分为带词性分词和不带词性分词，本文对文本分词的处理结果展示如表 2-1、表 2-2，代码详情见附件-数据预处理。

表 2-1 带词性分词

A eng市 n西湖 ns建筑 n集团 n占 v道 q施工 vn有 v安全隐患 iA eng市 n在水一方 i大厦 n人为 n烂尾 n多年 m安全隐患 i严重 aA3 eng区 n杜鹃 nr文苑 nr小区 n外 f的 uj非法 b汽车 n检测站 n要 v开业 n! x民工 n在 pA6 eng区明 nrfg发 v国际 n工地 n受伤 v工 n地方 n拒绝 v支付 v医疗费 nK8 eng县 n丁字街 nr的 uj商户 n乱 d摆摊 vK8 eng县 n南门 ns街 n干净 a整洁 a几天 m又 d是 v老 a样子 nK8 eng县 n冷 a江东 ns路 n蓝波 nr旺 a酒店 n外墙 n装修 v无人 n施工 vnK8 eng县 n九亿 m广场 n的 uj公厕 n要 v安装 v照明灯 nK4 eng县 n石期 n市镇 n老 a农贸市场 n旁边 f的 uj公厕 n旱 a厕 n里面 f脏乱差 lK eng市域 n轨道交通 n规划 n建议 n关于 pK eng市域 n轨道交通 n规划 n的

表 2-2 不带词性分词

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
A3 区 杜鹃 文苑 小区 外 的 非法 汽车 检测站 要 开业 ！
民工 在 A6 区明发 国际 工地 受伤 ， 工 地方 拒绝 支付 医疗费
K8 县 丁字街 的 商户 乱 摆摊
K8 县 南门 街 干净 整洁 几天 ， 又 是 老 样子
K8 县冷 江东 路 蓝波 旺 酒店 外墙 装修 无人 施工
K8 县 九亿 广场 的 公厕 要 安装 照明灯
K4 县 石期 市镇 老 农贸市场 旁边 的 公厕 （ 旱厕 ） 里面 脏乱差
K 市域 轨道交通 规划 建议
关于 K 市域 轨道交通 规划 的 建议
请问 A 市 乘坐 地跌 是否 可以 使用 “ 爱心卡 ”
地铁 5 号线 施工 导致 A 市锦楚 国际 星城 小区 三期 一个月 停电 10 来次

A6 区润 和 紫 郡 用电 的 问题 能 不能 解决

2.3 去除停用词

为提高文本利用率，需要去除文本中出现频率很高、但实际意义又不大的词。这一类主要包括了语气助词、副词、介词、连词等，通常自身并无明确意义，只有将其放入一个完整的句子中才有一定作用的词语。本文先构建停用词表，然后对文本进行去除停用词操作。

2.4 去除特殊字符

在去除停用词后，我们将对特殊字符，如：“ ” 、“，” 等标点符号或者空格进行去除，进一步提高文本挖掘的效率。去除特殊字符后的部分结果展示如表 2-3，代码详情见附件-数据预处理：

表 2-3 去除文本中的特殊字符

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
A3 区杜鹃文苑小区外的非法汽车检测站要开业！
民工在 A6 区明发国际工地受伤，工地方拒绝支付医疗费
K8 县丁字街的商户乱摆摊
K8 县南门街干净整洁几天，又是老样子
K8 县冷江东路蓝波旺酒店外墙装修无人施工
K8 县九亿广场的公厕要安装照明灯
K4 县石期市镇老农贸市场旁边的公厕（旱厕）里面脏乱差
K 市域轨道交通规划建议
关于 K 市域轨道交通规划的建议

2.5 文本向量表示

大部分文本分类方法都是基于空间向量模型的 ，在构建文本分类模型前，需将每一个文本表示成向量空间的向量，以每一个特征向对应一个维度，而每一个维度的值就是对应特征项在文本中的权重。

本文对文本向量表示的部分实现结果展示如表 2-4、表 2-5：

表 2-4 建立字典

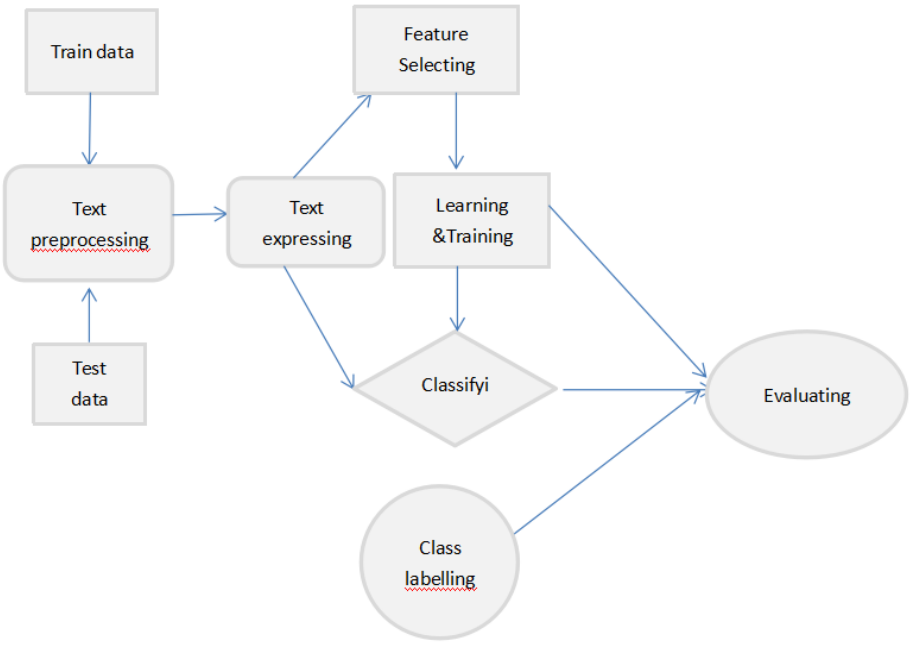
{'我们': 1, '领导': 2, '没有': 3, '西地省': 4, '现在': 5, '部门': 6, '请问': 7, '您好': 8, '希望': 9, '尊敬': 10, '10': 11, '一个': 12, '严重': 13, '学校': 14, '问题': 15, '政府': 16, '相关': 17, '工作': 18, '为什么': 19, '可以': 20, '老百姓': 21, '12': 22, '谢谢': 23, '解决': 24, '小区': 25, '居民': 26, '学生': 27, '人员': 28, '政策': 29, '教育局': 30, '公司': 31, '不能': 32, '厅长': 33, '什么': 34, '单位': 35, '办理': 36, '工资': 37, '本人': 38, '国家': 39, '有关': 40, '规定': 41, '业主': 42, '他们': 43, '生活': 44, '这样': 45, '一直': 46, '局长': 47, '是否': 48, '11': 49, '投诉': 50, '一下': 51, '医院': 52, '教育': 53, '情况': 54, '教师': 55, '多次': 56, '2015': 57, '30': 58, '老师': 59, '要求': 60, '2017': 61, '社保': 62, '2018': 63, '已经': 64, '小孩': 65, '处理': 66, '这个': 67, '2016': 68, '退休': 69'20': 70, '回复': 71, '孩子': 72, '收费': 73,

表 2-5 文本向量表示

[[1158, 571, 1313, 237, 1138, 881, 413, 1828, 982, 384, 671, 267, 104, 179, 882, 1185, 1778, 102, 410, 983, 468, 970, 307, 155, 1423, 67], [1030, 1138, 385, 725, 104, 411, 1732, 296, 1314, 258, 121, 66, 6, 690, 40, 167, 1139], [42, 205, 73, 25, 1079, 356, 609, 4, 610, 17, 161, 1121, 177, 1523, 41, 402, 170, 553, 294, 1895, 1605, 139, 1237, 579, 261, 408, 97, 309, 310, 101, 487, 424], [609, 1005, 323, 1, 488, 403, 534, 25, 271, 672, 112, 249, 121, 13, 2, 116, 450, 44, 16, 302, 1315, 103, 1484, 5, 3, 12], [609, 1005, 323, 1, 488, 403, 534, 25, 271, 672, 112, 249, 121, 13, 2, 116, 450, 44, 16, 302, 1315, 103, 1484, 5, 3, 12]]

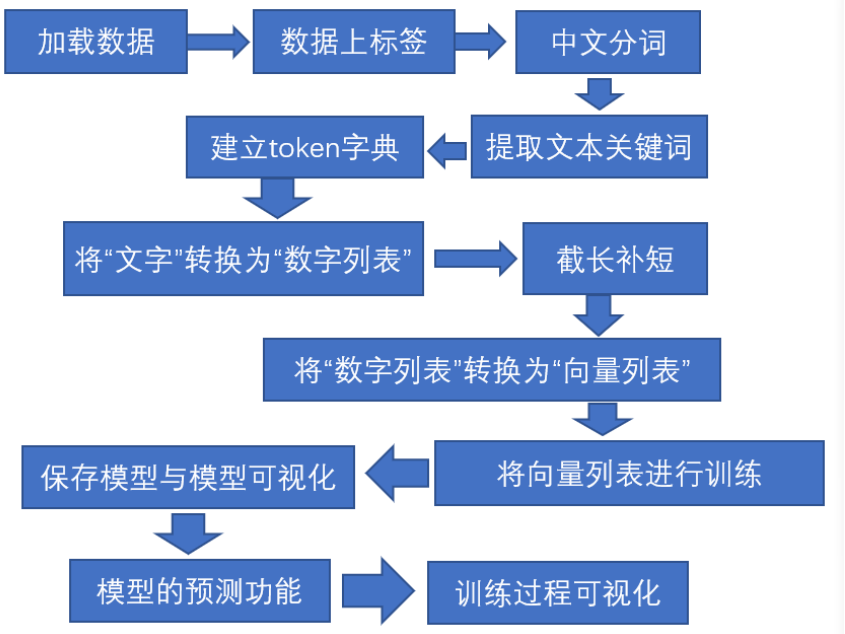
3 群众留言分类

文本分类属于自然语言处理领域，在机器学习领域，分类是在有标注的预定义类别体系下进行，从计算机角度看，是指计算机将载有信息的一篇文本映射到预先给定的某一类别或某几类别主题的过程。在预定义分类体系下，根据文本的内容相关性自动地判定文本与类别之间的关联。从数学角度来看，文本分类是一个函数映射过程，它将未标明类别的文本映射到预定义的类别，该映射可以是一一映射，也可以是一对多的映射，因为通常一篇文本可以同时关联到多个类别，文本分类的一般流程如图所示：



3.1 构建 CNN 分类模型

本文将根据附件 2 数据集构建关于文本留言内容的一级标签分类模型，实现流程如下：



本文共对模型进行了 5 次训练，训练结果如表 3-1、表 3-2、图 3-1 和图 3-2，代码详情见附件-分类模型构建。

表 3-1 输出

Layer (type)	Output Shape	Param #

embedding_1 (Embedding)	(None, 50, 32)	64000
conv1d_1 (Conv1D)	(None, 50, 256)	24832
max_pooling1d_1 (MaxPooling1	(None, 17, 256)	0
conv1d_2 (Conv1D)	(None, 17, 32)	24608
flatten_1 (Flatten)	(None, 544)	0
dropout_1 (Dropout)	(None, 544)	0
batch_normalization_1 (Batch	(None, 544)	2176
dense_1 (Dense)	(None, 256)	139520
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2570

Total params: 257,706、Trainable params: 256,618、Non-trainable params: 1,088

表 3-2 训练精确度

```
Train on 7368 samples, validate on 1842 samples
Epoch 1/5
7368/7368 [==============================] - 10s 1ms/step - loss: 1.7969 - accuracy:
0.3080 - val_loss: 2.7338 - val_accuracy: 0.0000e+00
Epoch 2/5
7368/7368 [==============================] - 5s 617us/step - loss: 0.9263 - accuracy:
0.6592 - val_loss: 2.9979 - val_accuracy: 0.0000e+00
Epoch 3/5
7368/7368 [==============================] - 5s 616us/step - loss: 0.4682 - accuracy:
0.8360 - val_loss: 3.1710 - val_accuracy: 0.0000e+00
Epoch 4/5
7368/7368 [==============================] - 5s 639us/step - loss: 0.2863 - accuracy:
0.9046 - val_loss: 3.3309 - val_accuracy: 0.0000e+00
Epoch 5/5
7368/7368 [==============================] - 5s 622us/step - loss: 0.1812 - accuracy:
0.9435 - val_loss: 3.5609 - val_accuracy: 0.0000e+00
```

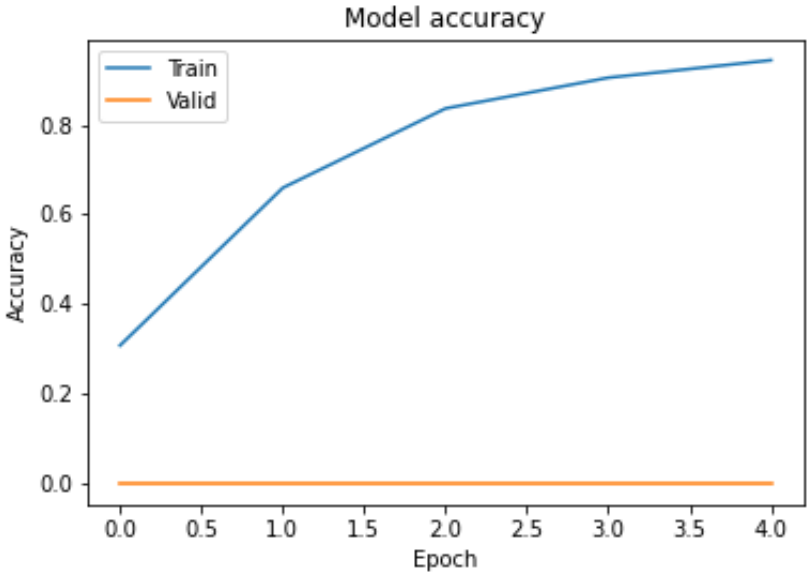


图 3-1 精确率变化曲线图

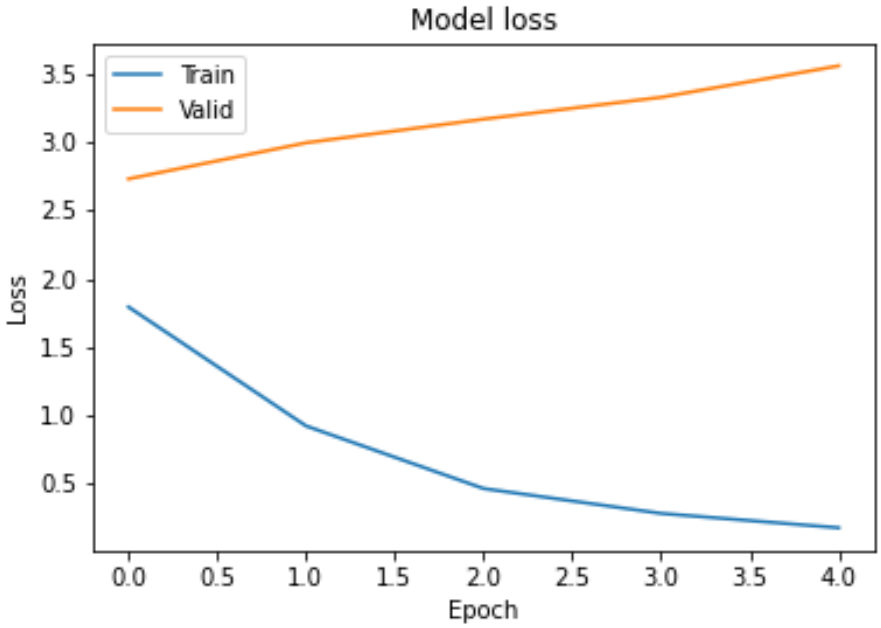


图 3-2 损失率变化曲线图

结果分析：由表 3-2 可知，随着训练次数的的增加，精确度不断提升，第五次训练精确度达到 94%，说明该分类模型的效果比较好。

3.2 构建 KNN 分类模型

3.2.1 背景说明

基于统计的机器学习方法已经成为自然语言研究领域里面的主流研究方法。事实上无论是朴素贝叶斯分类模型，还是支持向量机分类模型，也都采用了统计的方式。文本分类算法中一种最典型的基于统计的分类模型就是 k 近邻(k-Nearest Neighbor，kNN)模型，是比较好的文本分类算法之一。

3.2.2 原理介绍

本文将通过构建 KNN 分类模型，对留言进行分类。KNN 算法是通过测量不同特征值之间的距离进行分类，其核心思想是：如果一个数据在特征空间中最相邻的 k 个数据中的大多数属于某一个类别，则该样本也属于这个类别（类似投票），并具有这个类别上样本的特性。通俗地说，对于给定的测试样本和基于某种度量距离的方式，通过最靠近的 k 个训练样本来预测当前样本的分类结果。KNN 算法具体操作步骤：

（1） 将训练集样本转化为向量空间模型表示形式并计算每一特征的权重；

（2）采用类似步骤 1 的方式转化未标注文档 d 并计算相应词组元素的权重；

（3）计算文档 d 与训练集样本中每一样本的距离(或相似度)；

（4）找出与文档 d 距离最小(或相似度最大)的 k 篇训练集文本；

（5）统计这个 k 篇训练集文本的类别属性，一般将文档 d 的类归为 k 中最多的样本类别，部分实现代码如表 3-3 所示：

表 3-3 KNN 模型构建代码

```
iris = load_iris()
data = iris['data']
iris_data=pd.DataFrame(data=data,columns=
['sepal_length','sepal_width','petal_length','petal_width'])
iris_data["Species"] = iris[ 'target']
iris_data = iris_data.loc[iris_data['Species'] != 0,:]
x,y = iris_data.iloc[:,0:-1],iris_data.iloc[:,-1]
train_data,test_data,train_target,test_target  =  train_test_split(x,y,test_size  =
0.2,stratify = y)
min_max_scaler = preprocessing.MinMaxScaler()
X_train = min_max_scaler.fit_transform(train_data.values)
```

```
X_test = min_max_scaler.transform(test_data.values)
model_KNN = neighbors.KNeighborsClassifier()
model_KNN.fit(X_train,train_target)
Pre_label = model_KNN.predict(X_test)
metrics.confusion_matrix(test_target,Pre_label)
```

3.3 F-Score 评价方法

3.3.1 方法介绍

对于二分类问题，可将样例根据其真实类别与学习器预测类别的组合划分为真正例、假正例、真反例、假反例四种情形，如表 3-4 所示，令 TP、FP、TN、FN 分别表示其对应的样例数，则显然有 TP+FP+TN+FN=样例总数。

表 3-4 真实类别与预测类别组合表

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

查准率，又叫准确率，缩写表示用 P。查准率是针对我们预测结果而言的，它表示的是预测为正的样例中有多少是真正的正样例。公式如下所示：

$$P = \frac{TP}{TP + FP}$$

查全率，又叫召回率，缩写表示用 R。查全率是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确。公式如下所示：

$$R = \frac{TP}{TP + FN}$$

F1 度量公式如下所示：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

对于评价模型的好坏，我们用 ROC 曲线和 AUC 值来进行评价。

ROC，其主要分析工具是一个画在二维平面上的曲线——ROC 曲线。平面的横坐标是 false positive rate(FPR)，纵坐标是 true positive rate(TPR)。对

某个分类而言，我们可以根据其在测试样本上的表现得到一个 TPR 和 FPR 点对。这样，此分类就可以映射成 ROC 平面上的一个点。调整分类时候使用的阈值，我们就可以得到一个经过(0, 0)，(1, 1)的曲线，这就是此分类的 ROC 曲线。

AUC 被定义为 ROC 曲线下的面积，显然这个面积的数值不会大于 1。又由于 ROC 曲线一般都处于 y=x 这条直线的上方,所以 AUC 的取值范围在 0.5 和 1 之间。使用 AUC 值作为评价标准是因为很多时候 ROC 曲线并不能清晰的说明哪个分类的效果更好，而作为一个数值，对应 AUC 更大的分类效果更好。

3.3.2 模型测试

本文利用 F-Score 评价方法对构建好的 KNN 分类模型进行了四次测试，代码详情见附件-分类模型构建，测试结果如下：

表 3-5

	TP	FN	FP	TN	精度
第一次	10	0	2	8	90%
第二次	8	2	0	10	90%
第三次	19	1	1	19	95%
第四次	24	1	1	24	96%

表 3-6

召回率	精确度	F1 度量
0.95	0.95	0.9500000000000001

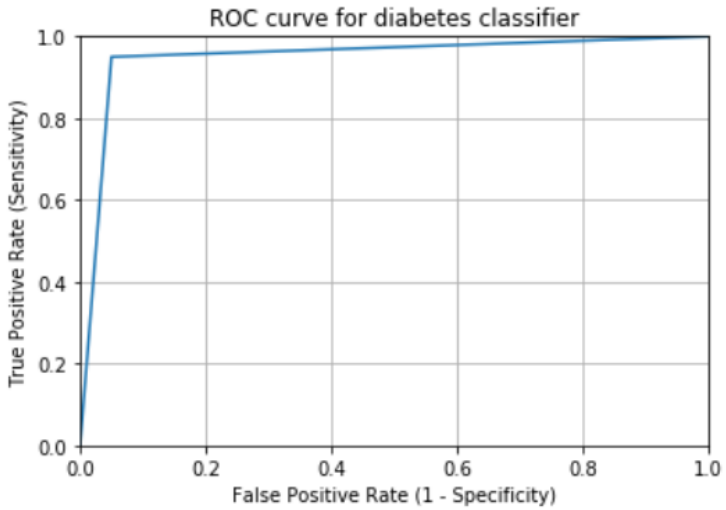


图 2-3

结果分析：由表 3-5 可知，模型训练次数达到 4 次时，精确度也提高到 96%，由图 2-3，AUC 值为 0.95，相比于 CNN 卷积神经网络分类模型，KNN 分类模型的精确度更高，由此可知利用 KNN 分类模型效果更好。

4 热点问题挖掘

4.1 含义解释

热点问题是当今社会许多人去关注的问题，当一个问题的关注度达到一定程度或者排名较前的时候，这个问题就是我们所说的热点问题，也是人们亟待解决的问题。

4.2 实验流程

首先对数据进行查看与分析，发现附件 3 的数据冗余，很多对于分析热点问题的数据是不需要的，这时候对于那些没有人关注的问题、零零碎碎的问题就不足以成为热点问题，所以可以对大部分数据进行筛选并且剔除冗余数据；其次把数据处理成可用于问题分析的格式，分别对数据进行清洗、去停用词、分词操作，停用词(Stop Word)是一类普遍纯在与文本中的常用词，并且脱离语境它们本身并不具有明显的意义，利用 jieba 库和正则表达式对数据进行相对应的分词清洗等工作；接着基于 word2vec 的训练模型，使用 gensim 自带的 word2vec 包进行词向量的训练，这个需要大量的词库，我们把找到的语料库用到我们的数据当中去，接着用 tag_id 标签、别名等操作，把词义相近的词

找出并进行文本的分类；最后按照点赞数和反对数的数量结合关键词词频对热点问题进行筛选，对筛选好的数据整理得到热点问题明细表（由于篇幅问题，热点问题详情表见附件-热点问题）以及热点问题表如表 4-1 所示：

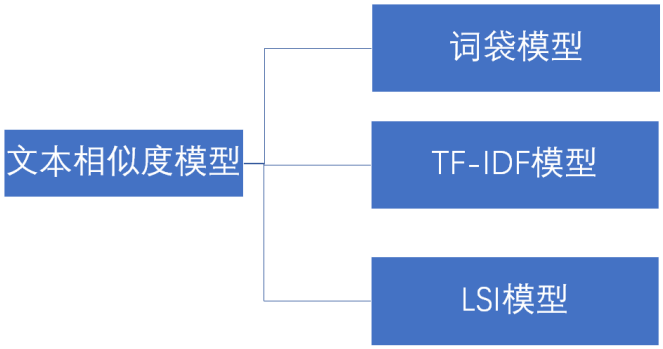
表 4-1 热点问题表

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	2358	2019/1/14 20:23:57-2019/5/28 15:08:51	A 市 58 车贷	A 市 58 车贷诈骗案件一直恶性循环得不到解决
2	2	2106	2019/3/22 22:35:30-2019/8/19 11:34:04	A 市 A5 区	小区物业保障和管理不当得不到解决
3	3	1768	2019/2/15 10:24:04—2019/4/11 21:02:44	A 市	入学难
4	4	672	2019/8/23 14:21:38-2019/9/5 13:06:55	A4 区绿地海外滩	小区与高铁之间的距离不合理
5	5	250	2019/1/13 10:13:09-2019/6/19 23:28:27	A 市	断水停水问题困扰据居民

5 答复意见的评价

5.1 文本相似度模型

实验流程图如下，分别构建三个模型计算各文本相似度，对比各个模型达到的效果，并根据得到的各文本的相似度对答复意见进行评价：



词袋模型：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，可以看成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。

TF-IDF 模型：一个词的权重由 TF * IDF 表示，其中 TF 表示词频，即一个词在这篇文本中出现的频率；IDF 表示逆文档频率，即一个词在所有文本中出现的频率倒数。因此，一个词在某文本中出现的越多，在其他文本中出现的越少，则这个词能很好地反映这篇文本的内容，权重就越大。

词频公式如下所示：

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

逆文档频率公式如下所示：

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

LSI 模型：LSI 是概率主题模型的一种，其核心思想是：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。

词袋模型部分结果如图 3-1 所示：

[[(0, 1), (1, 3), (2, 13), (3, 5), (4, 1), (5, 2), (6, 2), (7, 1), (8, 6), (9, 11), (10, 2), (11, 1), (12, 1), (13, 13), (14, 1), (15, 1), (16, 1), (17, 2), (18, 4), (19, 2), (20, 6), (21, 10), (22, 2), (23, 5), (24, 5), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 1), (33, 1), (34, 1), (35, 1), (36, 1), (37, 1), (38, 2), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (44, 3), (45, 1), (46, 2), (47, 1), (48, 1), (49, 1), (50, 1), (51, 1), (52, 1), (53, 1), (54, 1), (55, 1), (56, 1), (57, 1),

图 3-1

TF-IDF 模型运行结果：[(0, 0.15166861), (1, 0.2768673)]

结果分析：留言主题和留言详情的相似度为 15.17%左右，答复意见和留言详情相似度为 27.7%。

LSI 模型运行结果：[(0, 0.12758735), (1, 0.9918273)]

结果分析：留言主题和留言详情的相似度为 12.76%，答复意见和留言详情相似度为 99.18%

5.2 评价模型总结

使用 TF-IDF 模型来判断文章相似度存在一定的误差，TF-IDF 模型单纯以"词频"衡量一个词的重要性，并不够全面，因为有时重要的词出现的次数可能并不多。这种算法也无法体现词的位置信息，出现位置靠前的词与出现位置靠后的词，都被视为重要性相同，这是不太正确的。相对于 TF-IDF 模型，LSI 模型更适用于大数据文本，因此本文采用 LSI 模型的结果，由结果可知，答复意见文本和留言详情文本的相似度为 99.18%，表明答复在一定程度上是对应上留言内容的，即答复质量相对较高。

参考文献

[1]机器学习 周志华著.

[2]NLP 汉语自然语言处理原理与实践 郑捷著.

[3]神经网络和深度学习 Michael Nielsen 著.

[4]解读卷积神经网络 CNN_book_weixs 魏秀参著.

[5]基于文本挖掘的论坛热点问题时变分析 吴柳著.

[6]网络留言分类中贝叶斯复合算法的应用研究 马小龙著.

[7]用 python 进行自然语言处理 Steven Bird、Ewan Klein 等著.

[8]精通 Python 自然语言处理 Deepti Chopra Nisheeth Joshi Iti Mathur 著.

[9]从电子政务到智慧政务范式转变,关键问题及政府应对策略 陈玎、陈贵梧著.

[10]基于文本挖掘的网络热点舆论分析_以问题疫苗事件为例 刘宁著.