“智慧政务”中的文本挖掘应用

摘 要

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，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。因此运用文本分析和数据挖掘建立基于自然语言处理技术的智慧政务系统已经是趋势所向，这对提升政府的管理水平和施政效率具有重大作用。

此次数据挖掘中，我们首先使用 python 工具对各个附件进行数据预处理、分词，去停用词，加入自定义词典等操作。实现数据的优化，增加数据的可建模度。

其次，对各个问题进行针对性操作。对于问题一，我们将数据中所有数字映射为一个占位符（Placeholder）以达到降噪的目的。接着我们将数据集切分为训练集和测试集（9:1），使用 TF-IDF 算法提取每段留言数据的关键词，再利用 TF-IDF 得到所有词的 TF-IDF 权重向量。以此为基础使用 xgboost 模型训练训练集，通过 F1 分数（F1-score）,精确度（precision）,召回度（recall）,以及绘制混淆矩阵（confusion_matrix），AUC、ROC 曲线对测试集结果进行评价。由评价结果对 xgboost 参数进行调优，以建立更好的模型。

对于问题二，由刚开始的数据处理，利用 TF-IDF 得到所有词的 TF-IDF 权重向量，使用 k-means 对权重向量进行聚类,得到留言数据中相似留言进行计数汇总统计。之后基于 reddit 热点排序算法进行改进，将前面得到的计数个数作为一项热点指标，得到各留言的热点得分，进行排序后输出结果。

对于问题三，将留言与回复形成问答簇，将每个簇的词转化为 word2vec 词向量，分别计算问答簇的词语相似度（相关性），其次计算问答簇的海明距离（hamming_distance），以及留言与回复的时间的跨度，以两个星期（14 天）为分界点。由前面得到的信息，对问答簇进行质量评价。

关键词：中文分词；分类模型及其评价；k-means 聚类；reddit 热点评价；余弦相似度；海明距离

Text Mining Application in "Smart Government Affairs"

Abstract

In recent years, as the online political inquiry platform has gradually become an important channel for the government to understand public opinion, the amount of text data related to various social conditions and public opinion has been increasing, which has brought the work of relevant departments that used to manually divide messages and organize hotspots. Great challenge. Therefore, the use of text analysis and data mining to establish a smart government system based on natural language processing technology has become the trend, which has a major role in improving the government's management level and governance efficiency.

In this data mining, we first use the python tool to perform data preprocessing, word segmentation on each attachment, to stop words, add custom dictionary and other operations. Realize data optimization and increase data modelability.

Second, carry out targeted operations on various issues. For question one, we map all the numbers in the data into a placeholder to achieve the purpose of dimensionality reduction. Then we divide the data set into a training set and a test set （9: 1）, use the TF-IDF algorithm to extract the keywords of each piece of message data, and then use TF-IDF to obtain the TF-IDF weight vector of all words. Based on this, use the xgboost model to train the training set, and evaluate the test set results through F1 score （F1-score）, precision （precision）, recall （recall）, and drawing confusion matrix （confusion_matrix）, AUC, ROC curve . Based on the evaluation results, the xgboost parameters are tuned to build a better model.

For the second problem, from the beginning of data processing, use TF-IDF to get the TF-IDF weight vectors of all words, and use k-means to cluster the weight vectors to get the summary statistics of similar messages in the message data. Afterwards, it is improved based on the reddit hotspot sorting algorithm, using the count number obtained earlier as a hotspot indicator to obtain the hotspot score of each message, and output the result after sorting.

For question three, the message and reply are formed into a question and answer cluster, the words of each cluster are converted into word2vec word vectors, and the word similarity （relevance） of the question and answer cluster is calculated separately, and the Hamming distance of the question and answer cluster is calculated （hamming_distance） The span of the time between leaving a message and replying is divided into two weeks （14 days）. Based on the information previously obtained, the quality of the question and answer cluster is evaluated.

Keywords: Chinese word segmentation； classification model and evaluation； k-means clustering； reddit hotspot evaluation； word2vec cosine, Hamming distance

目录

1、挖掘目标

本次数据挖掘的目标是通过与各类社情民意相关的文本数据，利用 jieba 分词工具对文本数据分词，采用 TF-IDF 算法，多分类模型，k-means 算法，reddit 排序算法以及文本相似度算法，达到以下三个目标：

(1) 利用文本分词对非结构化的文本进行数据挖掘，根据所得结果切分数据，将数据转化为 TF-IDF 权值向量，建立多分类模型并对模型进行评价，从而解决数据庞大的群众留言分类问题。

(2) 利用文本分词和 k-means 聚类算法得到留言相同的个数，并为它赋予一定的权重。采用基于 reddit 热点排序算法的优化，得出各个文本的热点得分并进行排序，从而得到大量留言数据中的热点问题。

(3) 利用 jieba 中文文本分词对问答簇进行数据处理，计算各个问答簇内部的余弦距离和海明距离，并计算问答簇的时效性，由此建立回复文本质量评价方案，从而对政府的回复进行评分，便于各部门参考后提高回复质量。

2、分析方法与过程

总体流程图

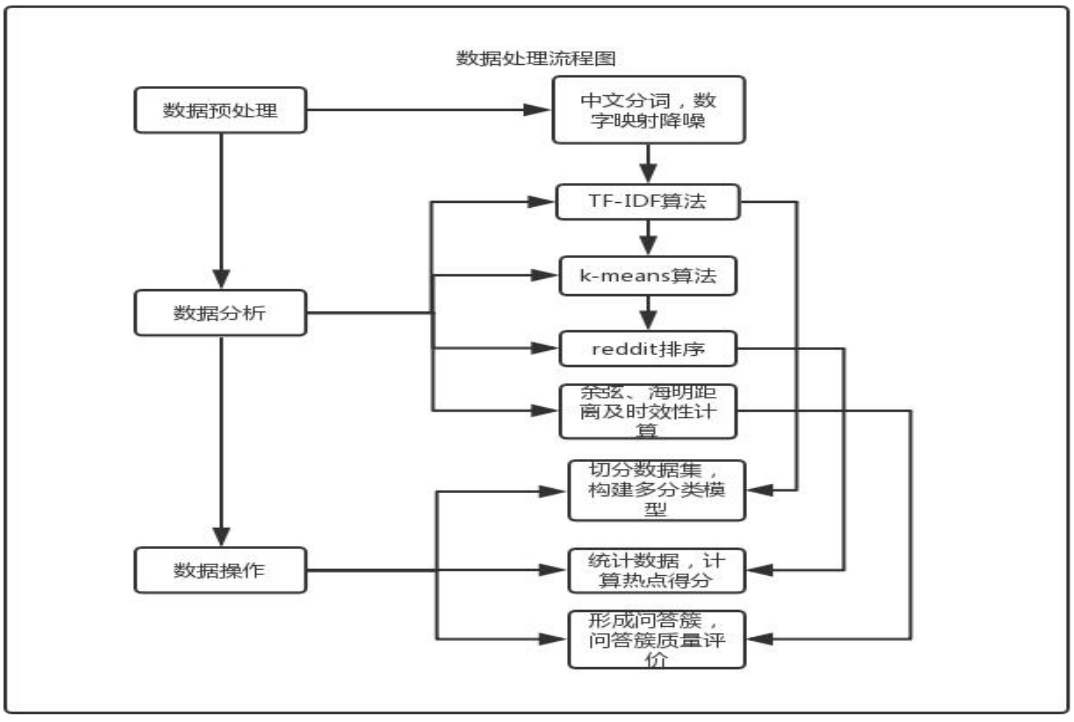


图 1：总体流程图

本用例主要包括如下步骤；

步骤一：数据预处理，对留言数据进行中文分词，将数据内容映射为同一占位符达到降噪效果。

步骤二：数据分析，使用 TF-IDF 算法将分词后的留言数据转化为权重向量，切

分数据集，建立多分类模型并评价模型效果。
步骤三：对附件三的留言详情内容采用 k-means 聚类算法进行分类。采用基于 reddit 热点排序算法的优化方法，得到各留言数据的热点得分。
步骤四：使留言与回复形成问答簇，对各个问答簇比较余弦相似度，海明距离，及时效性，建立关于回复质量的评价模型。

2.1 问题一分析方法与过程

2.1.1 问题一流程图

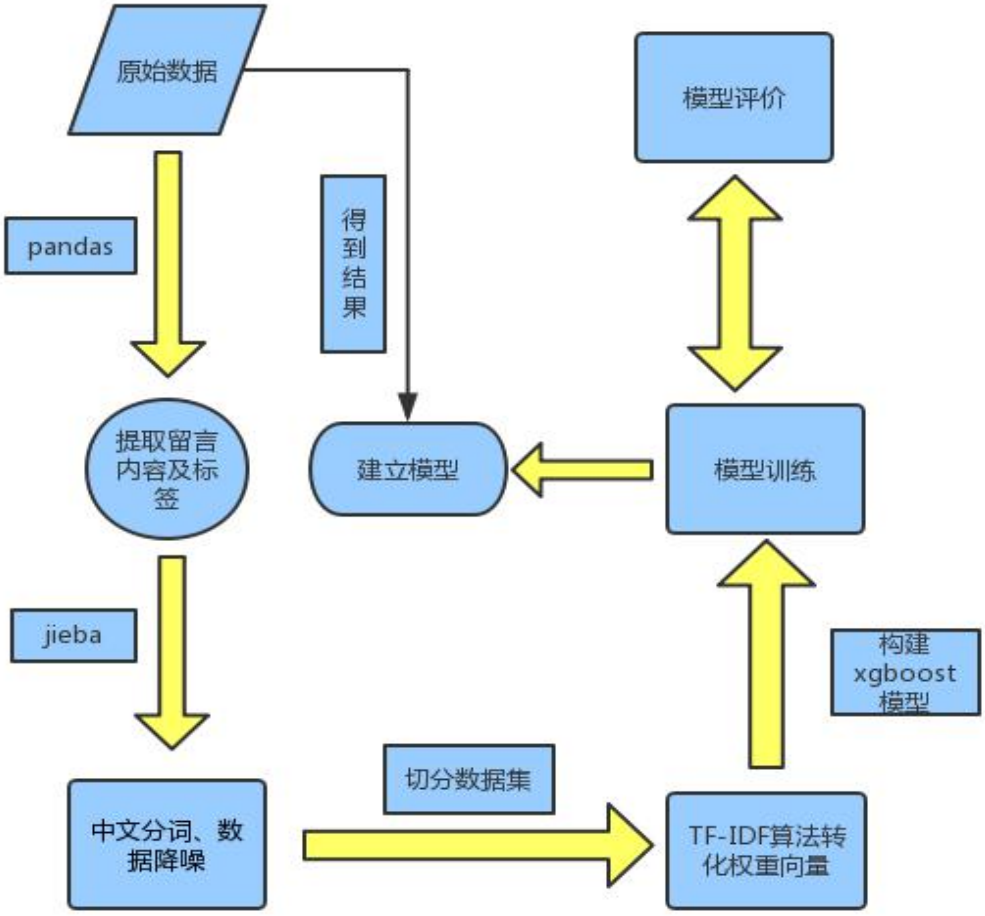


图 2：问题 1 流程图

2.1.2 数据预处理

2.1.2.1 提取数据

使用 python 中 pandas 包的 read_excel 方法提取 excel 中的留言数据及标签，并用 scikit-learn 中的 LabelEncoder 将文本标签（Text Label）转化为数字（Integer）。同时，为了减少数据中数字对后期处理的影响，将所有数字映

射为同一个符号（占位符），达到降噪的作用。之后，我们使用 scikit-learn 的 model_selection 模块中的 train_test_split 将数据集按 9:1 的比例切分成训练集（x_train,y_train）和测试集（x_valid,y_valid）。

2.1.2.2 中文分词及切分数据集

在构建模型之前，先要把非结构化的文本信息转化为计算机能识别的结构化信息，所以先要对附件中的留言数据进行中文分词。我们采用了 python 的中文分词包 jieba 进行分词，jieba 包使用了基于前缀词典而实现的高效词图扫描，生成了句子中所有可能成词可能性所构成的有向无环图（DAG），同时其运用了动态规划查找最大概率路径，找出基于词频的最大可能切分组合，对于未登录词，jieba 采用基于汉字成词能力的 HMM 模型，使得能更好地实现分词效果。

特别的，我们加入了停用词，以及自定义词。由于 jieba 分词中会把一些专有名词分开，（如“魅力之城”，jiaba 分词为“魅力”，“之”，“城”），因此加入自定义词典，能有效地提高分词效果。

2.1.2.3 TF-IDF 算法

对数据集分词后，需要将这些词语转换为计算机可以处理的向量，以便进行数据处理。我们采用 TF-IDF 算法，将留言数据转化为权重向量。TF-IDF 算法原理如下：

第一步：计算词频 TF（Term Frequency）。

$$词频(TF) = 某个词在文章中的出现次数$$

由于文章有长有短，为了便于不同文章的比较，进行"词频"标准化。

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

或者

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

第二步：计算逆文档频率 IDF（Inverse Document Frequency）

需要一个语料库（corpus），用来模拟语言的使用环境。

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

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）。IDF 越大，说明该

词在区分该内容的特征能力越强，即该词越能代表该文本。
第三步：计算 TF-IDF 值（Term Frequency - Inverse Document Frequency）

$$TF-IDF = 词频(TF) \times 逆文档频率(IDF)$$

因此 TF-IDF 与一个词在文档中的出现次数成正比，与该词在整个语料库中的出现次数成反比,TF-IDF 值越大,则该词越重要。计算 TF-IDF 值并进行排序,TF-IDF 值大的即为留言数据的关键词。

2.1.3 搭载多分类模型及初步评价

搭载模型前我们构建了模型评价方法——多分类对数损失函数（multiclass_logloss），以便对后期选择模型时提供判断依据。

（ps:对数损失越小，模型质量越高）

我们通过改写 python 中基于 TF-IDF 算法的 TfidfVectorizer 类得到 NumberNormalizingVectorizer 类，计算 TF-IDF 值并提取文本特征来 fit 训练集和测试集，将训练集（x_train,y_train）分别搭载到各个多分类模型中，所述模型包括逻辑回归（Logistic Regression），朴素贝叶斯（Naive Bayes），支持向量机（SVM），Xgboost 模型，基于 word2vec 词向量的 Xgboost 模型，深度学习（Deep Learning）等。（详情见附件 try1.py,try2.py。）
对于上述分类模型，我们得到初步对数损失数如下：

模型	逻辑回归	朴素贝叶斯	支持向量机	Xgboost	基于 TF-IDF-SVD 的 Xgboost 模型	基于 word2vec 词向量的 Xgboost 模型	深度学习
Logloss	0.567	0.637	0.362	0.401	0.392	0.709	0.672

表一：各分类模型初得分

由上表，我们决定选用基于 TF-IDF-SVD 的 Xgboost 模型（该模型较支持向量机具有优化空间），并在之后对其进行超参数优化，提升模型质量。

2.1.4 Xgboost 原理

XGBoost 目标函数定义为：

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss　　Complexity of the Trees

$$where\ \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

目标函数由两部分构成，第一部分用来衡量预测分数和真实分数的差距，另一部分则是正则化项。正则化项也包含两部分，其中 T 表示叶子结点的个数，w

表示叶子节点的分数。γ 可以控制叶子结点的个数，λ 可以控制叶子节点的分数超出，防止过拟合。

新生成的树用来拟合之前预测的残差的，即生成 t 棵树后，预测分数可以写成：

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

将目标函数改写成：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

基于此，我们应该去求一个 f_t 能够最优的目标函数。XGBoost 的想法是利用其在 f_t=0 处的泰勒二阶展开近似它。所以，目标函数近似为：

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

其中 g_i 为一阶导数，h_i 为二阶导数：

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

由于前 t-1 棵树的预测分数与 y 的残差对目标函数优化不影响，可以直接去掉。又每个样本都最终会落到一个叶子结点中，所以我们可以将所有同一个叶子结点的样本重组起来，过程如下图：

$$\begin{aligned} Obj^{(t)} &\simeq \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\ &= \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T \end{aligned}$$

由此，通过改写，我们将目标函数改写成关于叶子结点分数 w 的一个一元二次函数，求解最优的 w 和目标函数值就变成中学问题了，直接使用所学的顶点公式即可。最后所得最优的 w 和目标函数公式为：

$$w_j^* = -\frac{G_j}{H_j + \lambda} \qquad Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$$

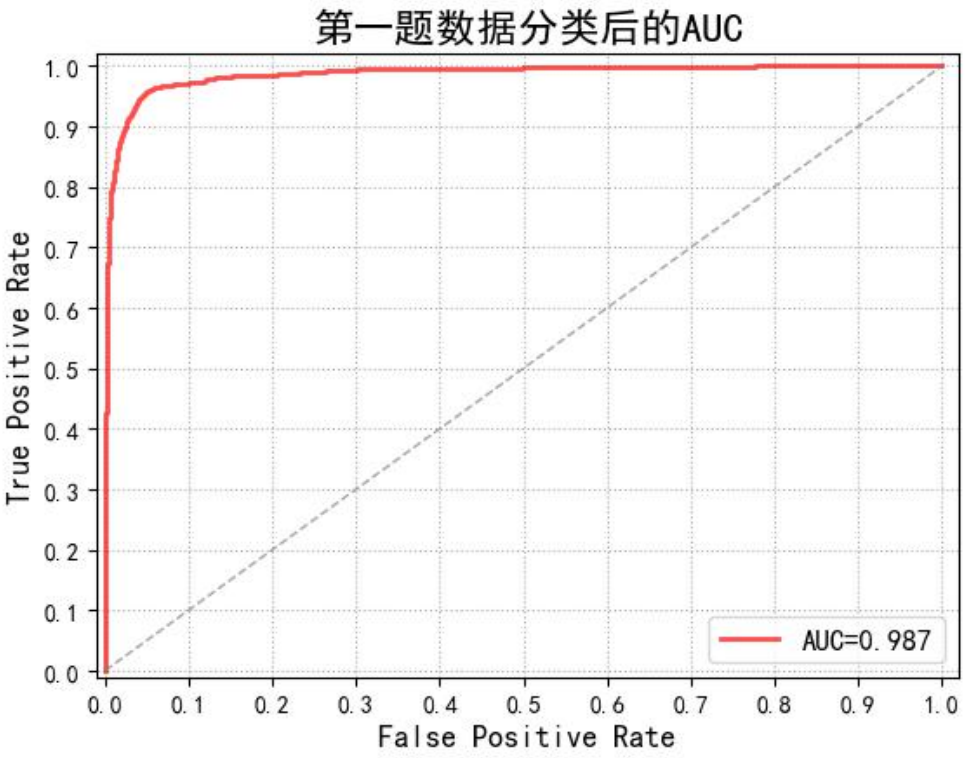
2.1.5 超参数优化与模型评价

我们使用网格搜索（GridSearch）对基于 TF-IDF-SVD 的 Xgboost 模型进行超参数优化。步骤如下：

（1）初始化模型默认参数。
（2）寻找最佳的 n_estimators（弱学习器的数量）。
（3）更新模型默认参数。
（4）对另一个参数调优。
（5）重复（3），（4），得到最近参数组合，优化结束。

超参数优化后我们加入了更多的模型评价方法，包括精确度（precision），召回率（recall）,f1-score,绘制 AUC 曲线，绘制混淆矩阵等，以全面评价模型质量（实现操作请见附件 plt.py）。

所得 AUC 曲线如下：



图三：AUC 曲线

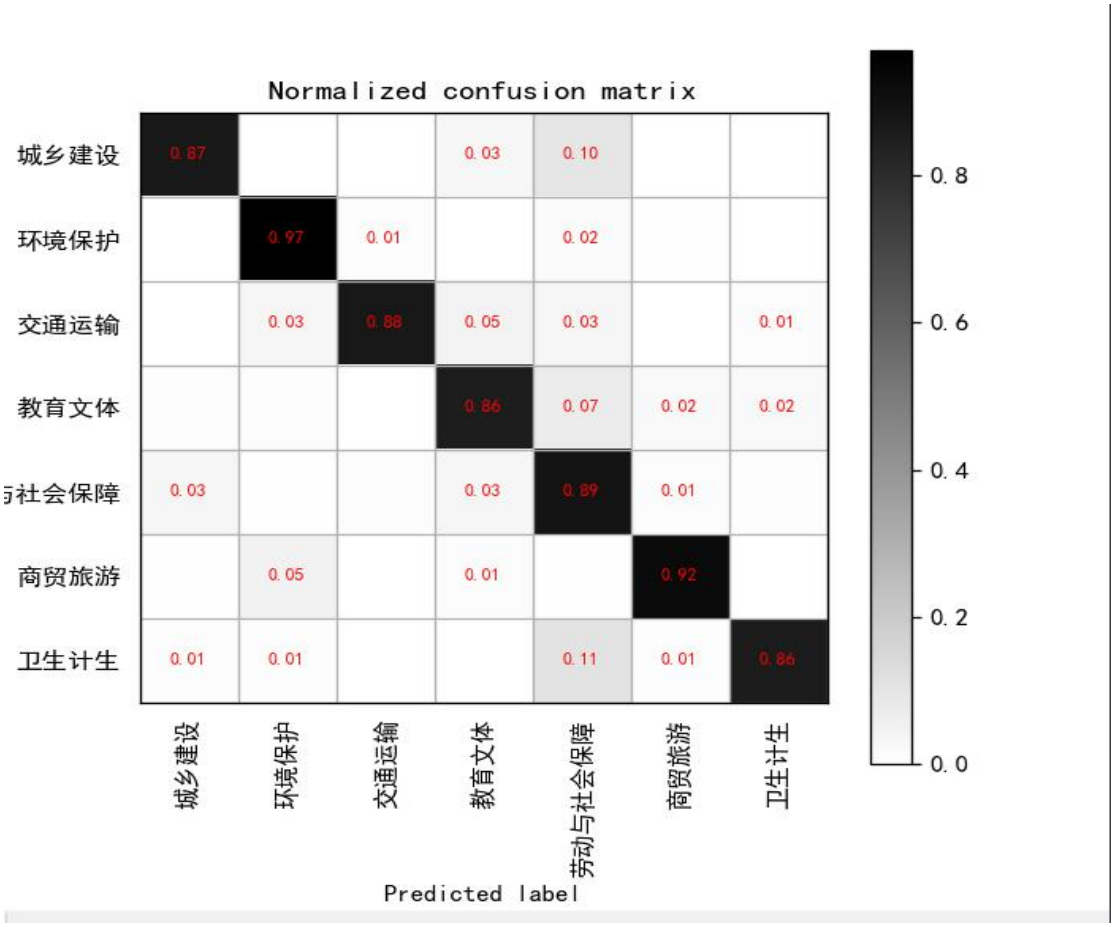


图 4：混淆矩阵图

2.1.6 结果分析

由混淆矩阵图，可直观的看出该模型在所给数据上的表现具有较好的表现，且具有提升空间。我们将该模型保存下来，以便以后用于解决留言分类问题。至此，我们得到了该基于 TF-IDF-SVD 的 Xgboost 模型。（参见 tdb1.py） .

2.2 问题二分析方法与过程

2.2.1 问题二流程图

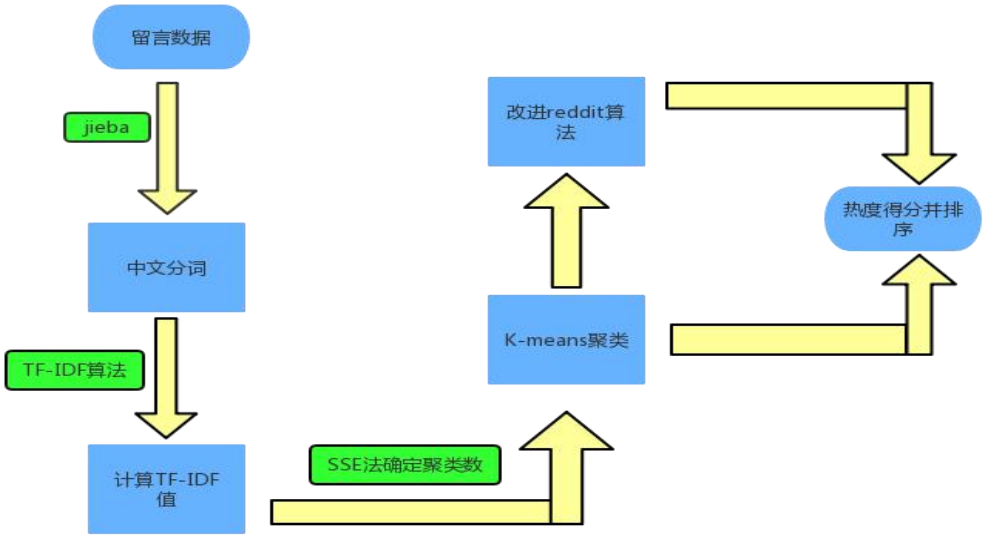


图 2 问题二流程图

2.2.2 留言数据聚类

如题一，对留言数据进行中文分词后计算 TF-IDF 值，再使用 K-means 聚类算法进行聚类。将附件 3 另存为附件 3 另.csv，便于后期将数据写入。

K-means 聚类算法的处理流程如下：

(1) 通过 SSE 方法，选择最佳 K 值。

(2) 根据第一步确定的 K 值，进行聚类。

(3) 通过轮廓系数检验聚类效果，范围在[-1,1],一般越接近 1，聚类效果越好，并将所属类别写入附件 3 另.csv

2.2.2.1 确定文本聚类数

对于所给数据，我们发现其中有重复数据，考虑将其去重，但因为该数据为群众留言数据，且重复留言并不是在同一天所发表，去除掉恶意刷热度的可能，因此我们不对原始数据去重，对其进行聚类。在使用 k-means 聚类算法前，要先确定 K 值（聚类类别总数），我们采用了手肘（SSE）法来确定 K 值。SSE 其实是一个严格的坐标下降（Coordinate Decendet）过程。设目标函数 SSE 如下：

$$SSE\left(c_1, c_2, \cdots, c_k\right) = \sum(x-c)^2$$

采用欧式距离作为变量之间的聚类函数。每次朝一个变量c_i的方向找到最优解，也就是求各个偏导数，然后等于 0，可得 c_i= （1/m）∑x（m 是 c_i 所在的簇的元素的个数）也就是当前聚类的均值就是当前方向的最优解（最小值）。这样可以保证 SSE 每一次迭代时，都会减小，最终使 SSE 收敛。由于 SSE 是一个非凸函数（non-convex function），所以 SSE 不能保证找到全局最优解，只能确保局部最优解。但是重复执行几次 kmeans，选取 SSE 最小的一次作为最终的聚类结果。eg:下图中 k 值最佳选择为 3

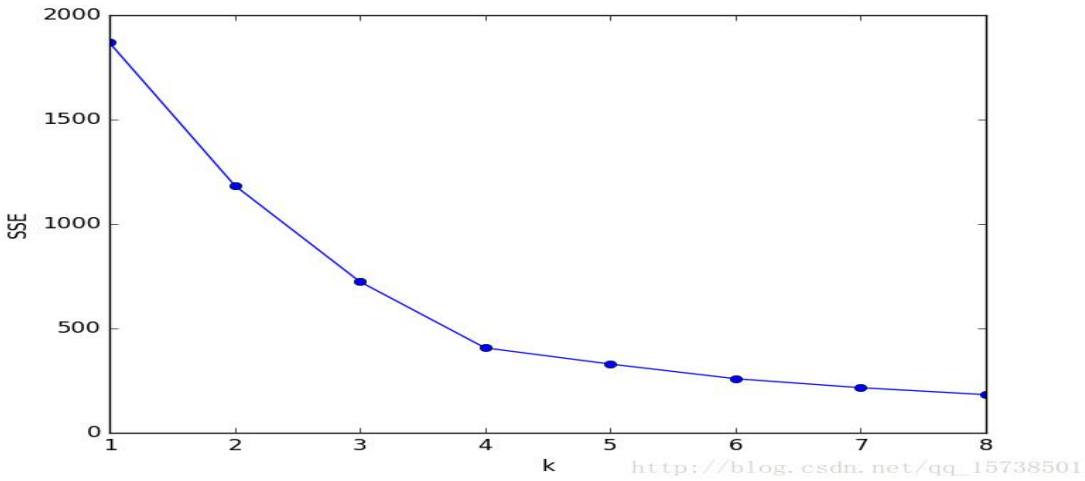


图 3 手肘法选择 K 值

由此，我们使用上述方法确定了相对适合附件三的 K 值，具体 python 操作见附件 k_determine.py。

2.2.2.2 K-means 聚类

K-means 聚类算法的原理为：对于给定的样本集，按照样本之间的距离大小，将样本集划分为 K 个簇。让簇内的点尽量紧密地连在一起，而让簇间的距离尽量地大。

如果用数据表达式表示，假设簇划分为（C1,C2,...Ck），则我们的目标是使平方误差 E 最小（最优解）：

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|_2^2$$

其中 μ i 是簇 Ci 的均值向量，也称为质心，表达式为：

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

如果我们想直接求上式目标函数的最优解并不容易，这是一个较难的问题，需要使用最小二乘法和拉格朗日原理。因此采用启发式的迭代方法。K-Means 采用的启发式方式很简单，用下面一组图就可以形象地描述。

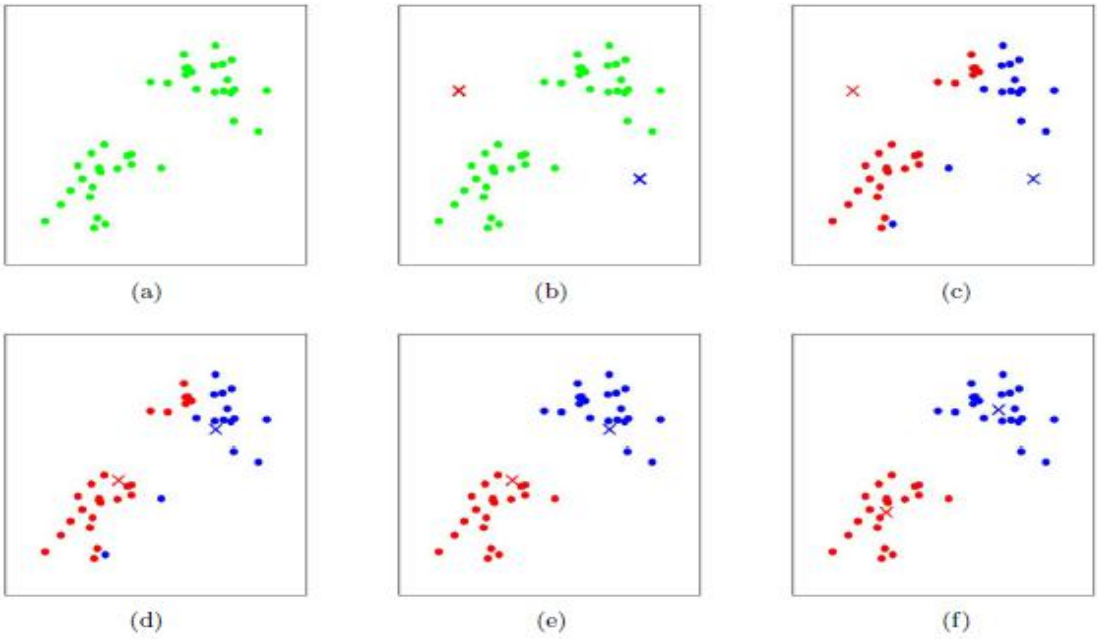


图 4 K-means 算法解析图

上图 a 为初始的数据集，假设 k=2。图 b 中，我们选择了两个类别质心，即图中的红色和蓝色质心，然后分别求图中中所有点到这两个质心的距离，并标记每个样本的类别和该样本距离最小的质心的类别，如图 c 所示。接着，经过计算样本和红色质心和蓝色质心的距离，我们得到了所有样本点的第一轮迭代后的类别。

此时我们对我们当前标记为红色和蓝色的点分别求其新的质心，如图 4 所示，两质心位置已经发生了变动。图 e 和图 f 重复了我们在图 c 和图 d 的过程，即进行迭代，将所有点的类别标记为距离最近的质心的类别并求新的质心。最终我们得到的两个类别如图 f。

2.2.3 基于 reddit 算法的热度排名算法

2.2.3.1 reddit 排名算法原理

Reddit 排名算法数学描述如下：

Given the time the entry was posted A and the time of 7:46:43 a.m. December 8, 2005 B, we have t_s as their difference in seconds

$$t_s = A - B$$

and x as the difference between the number of up votes U and the number of down votes D

$$x = U - D$$

where $y \in \{-1, 0, 1\}$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and z as the maximal value, of the absolute value of x and 1

$$z = \begin{cases} |x| & \text{if } |x| \geq 1 \\ 1 & \text{if } |x| < 1 \end{cases}$$

we have the rating as a function $f(t_s, y, z)$

$$f(t_s, y, z) = \log_{10} z + \frac{yt_s}{45000}$$

图 5 reddit 算法原理

Reddit 话题排序算法总结:

1、提交时间是一个很重要的参数，通常新的话题将会高于老的话题的分数。

2、前十个投票和后 100 个投票的作用是相同的，举个例子，一个有 10 的“顶”话题和一个有 50 个“顶”的话题他们的排名是相同的。

3、得到支持票和反对票持平的争议话题和得到票大多为支持的话题相比排名将会较低。

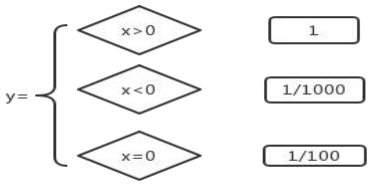
2.2.3.2 基于 reddit 算法改进的热度排名 Red sorted 算法

由于所给数据为群众留言数据，为防止数据被“反对数”淹没，我们对 reddit 算法进行改写。所述改写为：将下列数学表达式

where $y \in \{-1, 0, 1\}$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

改写为：



我们通过 K-means 聚类算法得到了各个文本所属类别及其类别所含文本数。由于热点问题为某一时段内群众集中反映的某一问题，聚类后所得各类别所含文本数的多少将直接影响问题的热点得分，因此我们为其赋予了一定的权重。其数学表达式为：

$$c=count^{5}。$$

其中，count 为聚类得到的与该文本同类的文本个数。

2.2.4 热度评价及其结果分析：

综上，我们得到了改进后的热度排名算法 Red sorted，定义相应的评价指标，从而得到评价结果。所述评价指标为：

$$f（t,y,z,c）=f（t,y,z）+c$$

注：f（t,y,z）为原上述函数

将附件 2 另存为附件 2.另 csv 后将热度得分，排序后按要求分别将得到热点问题表.xls 和热点问题留言明细表.xls。(具体 python 操作见附件 tdb2.py)。

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	26375.65771	2019/7/7 至2019/8/30	伊景园滨河苑购房	A市伊景园滨河苑捆绑销售车位
2	2	12742.77165	2019/4/30至2019/10/16	A市市民	A市泉星公园项目优化建议
3	3	10812.38281	2019/11/13至2020/01/25	A市丽发新城居民	社区附近搅拌站严重扰民
4	4	10593.43586	2019/06/12至2019/09/08	魅力之城小区居民	小区楼下夜宵店深夜经营严重扰民
5	5	10547.72384	2019/06/12至2019/09/08	诺亚山林小区居民	小区门口设立医院

表二：题二热点问题表

我们得到了如上的热点问题表，由此表可准确直观的看出群众留言中较受关注的热门问题。因此我们可以通过数据挖掘分析得到数据中的热点问题，为各相应部门提供参考标准，解决人工筛选中存在的费时费力问题。

2.3 问题三分析方法与过程

2.3.1 问题三流程图

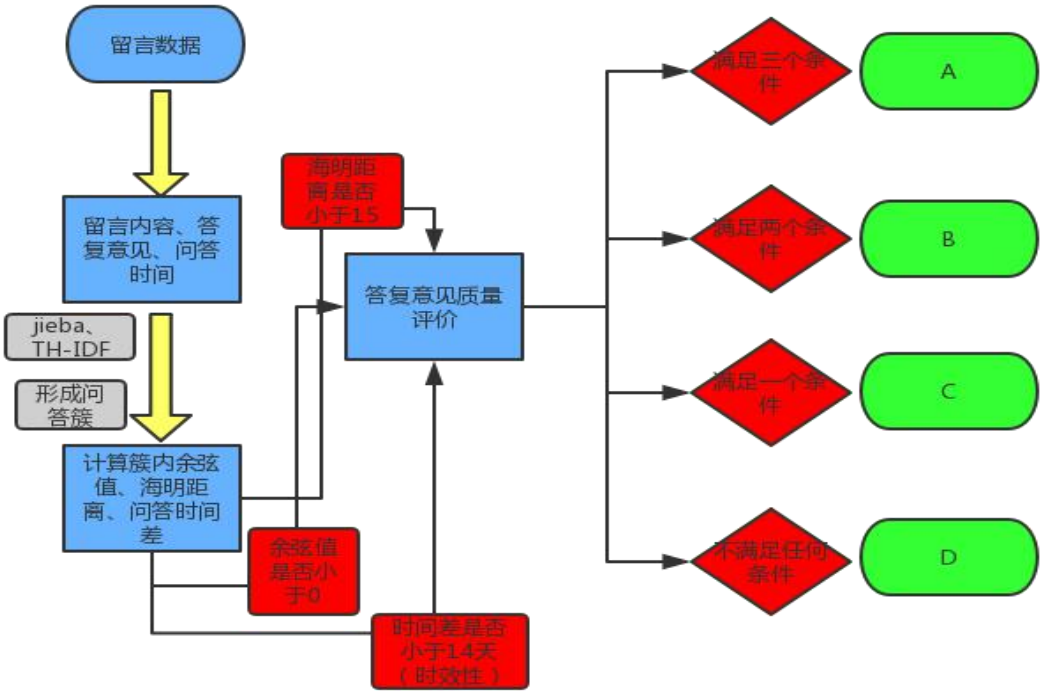


图 6 问题三流程图

2.3.2 数据预处理

如前题的操作，我们将留言数据中的留言内容和答复意见提取出来，特别的，我们将之作为问答簇，以便后面对各个簇进行分析。接着，我们对留言和答复时间进行截取，只留下时间的年月日，忽略分、秒等，减少计算量。

2.3.3 答复质量评价指标量化

我们将评价指标定义为三个标准：相关性、完整性、时效性，并分别使用余弦相似度，海明距离（hamming_distance）、问答时间差将其量化。于是，对于每个问答簇，我们计算其余弦相似度、海明距离，及问答时间差。由于余弦值越大，文本相似性更好，我们将余弦值大于 0 的问答簇记布尔值“1”，小于 0 的记为“0”。将海明距离小于 15 的记布尔值“1”，大于 15 的记为“0”。考虑到数据为群众留言与政府答复文本，我们以两个星期作为问答时间差判断节点。将时间差小于 14 天的记布尔值“1”，大于 14 的记为“0”。在 python 操作中将结果存放至三个列表中，具体见附件 tdb3.py。

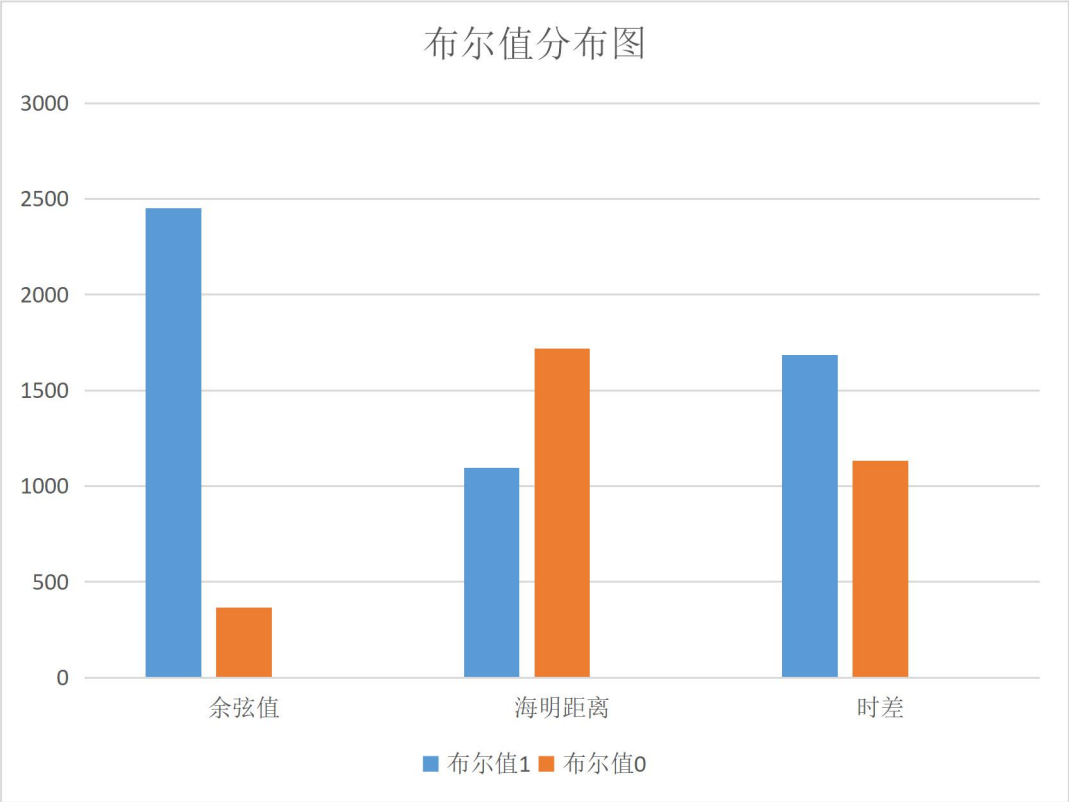


图 7：各条件布尔值分布图

由此，我们将遍历三个存放布尔值的列表，并设定对应评价指标。所述评价指标为：

（1） 若问答簇满足三个条件（布尔值均为 1），则记为 A。

（2） 若问答簇满足两个个条件（布尔值为两个 1，一个 0），则记为 B。

（3） 若问答簇满足一个条件（布尔值为一个 1，两个 0），则记为 C。

（4） 若问答簇满足零个条件（布尔值均为 0），则记为 D。

另存附件四为附件四另.csv 并将所得结果写入。其中质量等级 A 为优秀，B 为良好，C 为及格，D 为不及格。

2.3.4 结果与分析

结果	举例		
留言编号	30019	12031	12163
留言用户	UU008151	UU008219	UU0082207
留言主题	在 A6 区准备全款购买二手房事项的咨询	希望延长城乡公交 2 路车（A 市晚报-黄花镇）下班时间	请求 701 或 915 路公交调整经过 A 市大学
答复意见	已收悉	网友：您好！留言已收悉	网友：您好！留言已收悉
留言时间	2016/11/3 10:00:17	2014/8/23 18:52:08	2014/7/9 13:43:44
答复时间	2016/11/22 12:25:56	2014/9/4 16:30:25	2014/8/13 17:43:11

评价等级	D	C	D

表三：评价结果举例

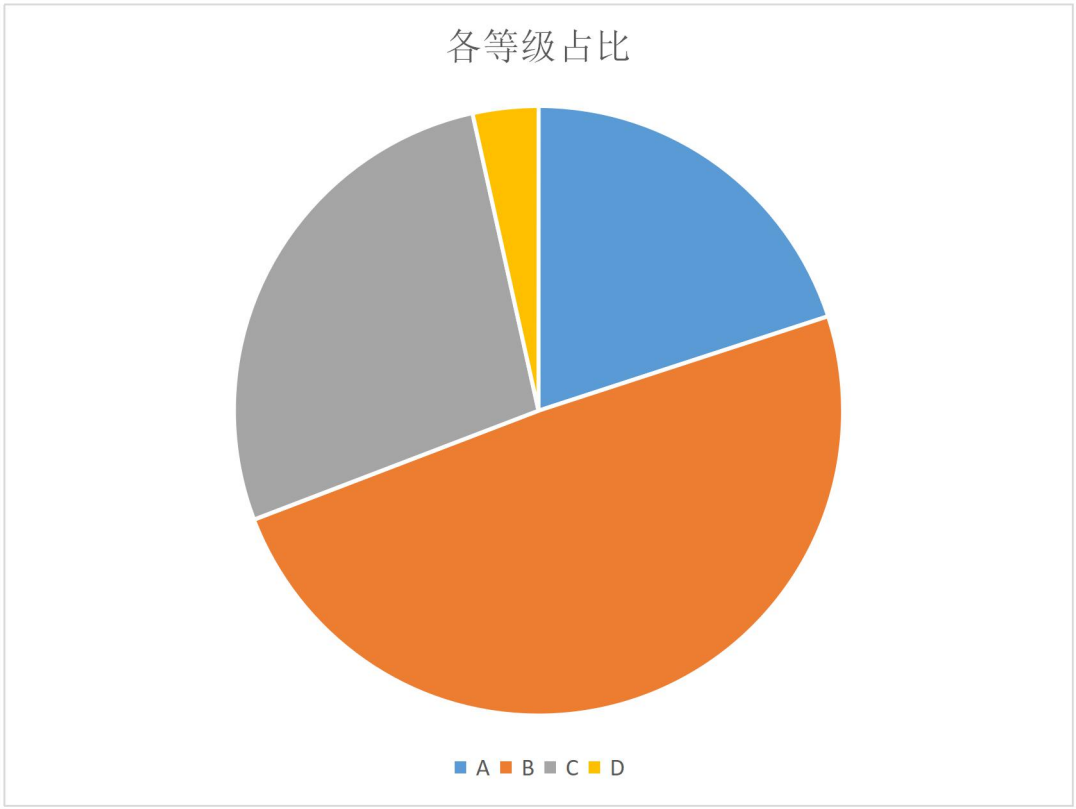


图 7：各等级数量

由于是对答复意见的质量评价，我们将侧重点放在低等级质量上。由上图，我们可以了解到，在对群众留言回复时，我们存在些许问题，导致答复质量不高。因此我们应该在一定的时间内做出相应答复，并且尽量不应以已收悉等较为无实际意义的词为回复，应该基于留言问题，做出相对真实可行的回复。群众的留言问题是否能够解决是衡量一个职能部门能力的标准，通过数据分析得到的答复质量等级评价能作为一个较好的参考，帮助各部门提高答复意见质量。

3、结论

对群众留言数据进行分类，提取热点问题，以及对答复进行评价等问题在当今数据飞速产生的世界已成为燃眉之急。传统的人工处理将耗费大量的人力物力，使用数据挖掘和文本处理技术将大大地改善这样的情况。

本文使用了基于 TF-IDF-SVD 的 Xgboost 多分类模型，将数据集切割为训练集和测试集，使用多种模型评价方法，最终建立了较优秀的多分类模型，以用于解决留言数据分类。

使用 K-means 聚类算法及基于 reddit 热点排序算法，建立相依的热度评价指标，由数据分析得到热度排名前五的留言，进而使相应部门重视处理该热点问题，减少相应部门的工作量。

使用余弦相似度算法、海明算法、时效性计算，得出各个回复的质量等级。由分析结果可知，答复意见中仍存在一定比例的欠合格答复。所以，各相应部门应该查找原因，在处理群众留言时应该注意答复的质量和时效性。

4、参考文献

[1] 程学旗,杜慧,伍大勇,张瑾等.一种面向网络话题的热度评价方法.中国科学院计算技术研究所.2015
[2] 王千，王成，冯振元，叶金凤.K-means 聚类算法研究综述.2012
[3] 卢美莲,刘金亮,叶小卫,王明华,曹一鸣,李佳珊.基于新闻内容和主题特征的个性化新闻推荐装置和方法.北京邮电大学.2012
[4] 王娟,席天为,赵克全.基于改进 K-means 算法的评价问题研究.重庆师范大学.2020
[5] 苏东出.基于 TF-IDF 和余弦相似度的图书馆 OPAC 系统地研究和实现.2019
[6] 方高林，刘怀全，郑全战.一种问答对的质量评价方法与系统.2009
[7] 李明兰.基于非固定长度散列表的无监督式海明距离搜索.青岛大学.2020
[8] 胡泽.在线问诊服务回答质量评价方法研究.哈尔滨工业大学.2019
[9] 灰灰.一文读懂机器学习大杀器 XGBoost 原理.知乎.2019