**《信息分析》课程作业**

**——线性降维与聚类分析**

**专业：** 信息管理学院 **学号：** 211820073 **姓名：** 胡涂

**专业：** 信息管理学院 **学号：** 211840182 **姓名：** 邓雨茵

**专业：** 信息管理学院 **学号：** 211820274 **姓名：** 郑宇

# 基于K-means的视频垃圾评论识别与消减方法

**摘要：**内容文字用楷体，五号字

**关键词：**K-means、垃圾评论识别、自然语言处理

## 前言

在当今互联网时代，评论已成为人们分享观点和思想的重要渠道。评论的存在不仅可以促进作者与读者之间的交流，还可以帮助作者改进作品，提高阅读者的体验。然而，与评论相关的垃圾评论问题也逐渐引起了人们的担忧。

尤其是对于视频平台，评论更是对于平台与作者的一个重要资源。精心设计的评论不仅可以吸引观众留言，而且可以提高制作者与平台的曝光度。但是大量的垃圾评论会降低评论的质量，影响观众的阅读体验和作者的自我成就感。同时大量的垃圾评论无助于视频作者改善视频质量，增长视频效益，并且诸如刷屏、灌水、恶意攻击等各种垃圾评大大降低了人们对评论信息质量和阅读体验的要求，甚至制约了视频APP平台的用户黏性。

在评论人工标注的过程中，常常存在主观性和不一致性的问题。而在垃圾评论标注这一任务中，人工标注常常会将本应是正常的评论标注成垃圾评论。因此，我们需要一种有效的方法来识别和过滤掉垃圾评论。聚类算法，尤其是K-means聚类算法，是一种非常有效的方法来处理分类问题，尤其是对于评论文本的分类问题。通过使用聚类算法，可以更加有效地自动识别垃圾评论并减小标注误差。

在本文中，我们将详细分析这种基于聚类的方法，对其有效性进行深入研究，并依据具体的视频评论进行实践研究。通过本文的研究，我们将选择出垃圾评论所具有的一些重要特征，并提供一个有助于解决视频垃圾评论标注问题的有效工具，同时提高视频平台的评论质量，提高作者和平台的普及性和用户满意度，以实现更强大的社区和更友好的交流环境。

## 数据集及特征构建

### 数据集介绍

数据来自Kaggle平台的Youtube Videos and the Comments(<https://www.kaggle.com/datasets/japkeeratsingh/youtube-videos-and-the-comments/code>)。 共379032条记录，5项特征。

数据字典如下：

|  |  |
| --- | --- |
| 特征名 | 描述 |
| User | 视频作者 |
| Video Title | 视频标题 |
| Video Description | 视频描述 |
| Video ID | 视频ID |
| Comment (Displayed) | 展示评论 |
| Comment (Actual) | 真实评论 |
| Comment Author | 评论作者 |
| Comment Author Channel ID | 评论者频道ID |
| Comment Time | 评论时间 |

在数据集中，并没有提供可供分析的特征以及相应的标签，所以特征需要人工进行构建，并进行无监督学习以达到识别与标注目的。

### 描述性特征构建

通过资料的收集，我们认为垃圾评论主要具有同一用户多次重复、情感极性偏向、评论价值较低、长度较短、大写字母占比较多的特征。

###### 2.1 同一用户多次重复

用户在不同视频下的评论总数。这类评论单独看起来正常,但是从整体来看,会发现不同视频之间含有同一评论者发表的相同或者相似的评论,称为重复评论，此类评论的目的不是对视频进行评论，而是广而告之,属于垃圾评论。评论重复数越多，就越有可能属于垃圾评论。因为该数据集视频数量较少，故将用户在不同视频下的评论总数作为一个特征，认为评论越多越可能是垃圾评论的制造者。

###### 2.2 情感极性偏向

句子的情感极性。Jindal等人发现了评论者的一些行为很可疑,比如一般正常的评论者所发表的评论往往包括积极的、中性的和消极的评论，但是有一部分评论者发表的评论全部是积极的或者全部是消极的。经过试验发现这些行为特征与评论造假的关联性很高。因此我们利用Python的NLTK库对句子进行情感分析，选取积极程度和消极长度二者的最大值，称为情感极性，情感极性越大，该评论是垃圾评论的可能性越大。

（李耀鹏 & 徐德华.(2019).垃圾评论识别研究综述. 科技与创新(04),22-23+25. doi:10.15913/j.cnki.kjycx.2019.04.022.）

###### 2.3 评论价值较低

词的价值。正常的评论应表达出评论者的情感或者意见,而垃圾评论更多的则是描述一些信息,因此垃圾评论的名词数量较多，而正常的评论会更多使用动词，形容词等来充分表达自己的意见。名词度越高,，越有可能属于垃圾评论。定义为评论中包含的动词，形容词和限定词的数量除于评论的单词总数，用到的工具是Python的NLTK库的词性分析器。

###### 2.4 长度较短

评论的长度。通过统计实验发现,一般较短的评论是垃圾评论的可能性很大。

（李霄 & 丁晟春.(2013).垃圾商品评论信息的识别研究. 现代图书情报技术(01),63-68.）

###### 2.5 大写字母占比较多

大写字母占比。这一特征主要是针对英文评论，通过统计发现，大写字母较多的评论往往是垃圾评论，尤其是较短且全用大写字母时。

### 语义特征构建

###### 3.1 评论向量化

除了上述描述性特征构建，我们小组考虑了对评论源文本进行向量化以进行相似度比较。

垃圾评论在语义上可能并没有任何的相似程度，但是有价值的评论在语义上是较为相似的，在这种情况中，我们希望对文本进行语义上的比较，而不是仅仅比较单词或字符的相似性。因此，我们需要将文本转换为能够表示语义信息的向量。

而传统的词袋模型只关注文本中单词的个数，而不考虑单词之间的关系以及语义信息。

因此在语义向量化的任务中，我们采用BERT预训练模型进行向量化操作。

BERT-Whitening是一个使用预训练的语言模型BERT来进行文本表示的技术，以及针对该表示做出的一种嵌入学习策略。BERT是一种双向Transformer模型，它在预训练阶段使用了大量的无标签数据，将该数据输入模型并将模型调优，在此过程中模型可以构建出单词和句子的表示，并不仅仅是简单的词向量，同时包括了语义和上下文信息。然后，BERT-Whitening使用协方差矩阵和白化矩阵来处理BERT表示，使相似的文本在嵌入空间中更加接近，从而提高文本相似度计算的效果。

通过BERT-Whitening处理，将评论处理成了763维的向量。

###### 3.2 提取向量主成分

由于评论语义向量维数过高，选择利用PCA与方差选择的方法将其降维处理。由于本身语义向量的含义较难以解释，因此聚类结果并不依靠语义向量具体含义对其进行解释，仅凭借相似度判断，所以使用PCA降维所导致的解释性破坏可以忽略。

在处理流程中，首先对向量进行方差选择，降低PCA处理时的时间与空间开销。选择阈值设为0.1，最终选出了32维特征。

PCA方差解释占比选择为总方差的0.95，最后将向量降至3维。维数较低的同时能够较好的代表语义。

补充：BERT-Whitening这种预训练模型词向量化处理后进行PCA降维能够有效的提高各种任务的性能，具体不在此赘述。

### 特征构建完毕后的数据字典

|  |  |
| --- | --- |
| 特征名 | 描述 |
| Comment Author Counts | PCA1 & 2 & 3 |
| Polarity | 评论的情感极性 |
| Comment Length | 评论长度 |
| Worth Word Ratio | 有用词性占比 |
| PCA1 & 2 & 3 | 语义数值向量 |

## K-means及结果选择

### 基于不同的特征进行聚类

在开始聚类之前，为了便于判断依据各种特征聚类标注的效果与性能，我们选取了99条评论进行人工标注，将在聚类后依次查看各个聚类的混淆矩阵以进行判断。

同时，由于任务的原因，我们聚类的集群数目，即k值预先设置为2。

为了实现我们的目标——找出最能体现垃圾评论特征的指标，我们小组利用spss软件，在train数据集中选择对CommentAuthorCounts、Polarity、CommentLength、WorthWordRatio、PCA、前四个指标的联合特征、所有指标的联合特征分别进行多次K-means聚类将其聚为两类，并在所有聚类结果上打上标签：“1”代表有用评论，“2”代表垃圾评论。并将聚类后的结果与事先人工标注标签的test数据集进行对比，绘制出每个结果的混淆矩阵，并计算出精准率和召回率，通过混淆矩阵、精准率和召回率三个指标来选取最能体现垃圾评论特征的指标。

###### 1.1 归一化

不同评价指标（即特征向量中的不同特征）往往具有不同的量纲和量纲单位，这样的情况会影响到数据分析的结果，为了消除指标之间的量纲影响，需要进行数据标准化处理，以解决数据指标之间的可比性。在本项目中，我们使用最大最小标准化的方法，对CommentAuthorCounts和CommentLength两列数据进行了归一化处理，即

使得预处理的数据被限定在[0,1]内，消除奇异样本数据导致的不良影响，以及不同量纲所导致的聚类差异。

###### 1.2 初始聚类中心选择

基于我们小组对垃圾评论的定义及其带有的性质，我们对K-means聚类的初始聚类中心进行了抉择。

根据有用评论的稀缺性及垃圾评论的泛滥性，我们认为某个用户在该数据集中的评论次数越多，该用户越可能发表垃圾评论。故在初始聚类中心中将分类“1”中的CommentAuthorCounts的数值设为0，分类“2”中对应数值设为1。

同样，评论所带有的个人情感越强烈也即评论的情感极性越大，通常认为其越不理智，也即其为垃圾评论的可能性越大。故在初始聚类中心中将分类“1”中Polarity的数值设为0，分类“2”中对应数值设为1。

在我们的定义中，我们倾向于将类似于包含对视频内容的评价、不足的指出等内容的视频评论定义为有用评论，而这样的评论通常较长，且有用词性占比较高。故我们在初始聚类中心的设置中将分类“1”中的CommentLength和WorthWordRatio的数值均设置为1，分类“2”中的对应数值均设置为2。

最终的初始聚类中心如图所示：

|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| CommentAuthorCounts | 0 | 1 |
| Polarity | 0 | 1 |
| CommentLength | 1 | 0 |
| WorthWordRatio | 1 | 0 |
| CapitalLetterRatio | 0 | 1 |
| PCA1 | 0.74098169 | 3.99699697 |
| PCA2 | -0.26148267 | -0.13702365 |
| PCA3 | -0.08360202 | -0.4327849 |

### 聚类效果的评判标准

对于聚类标注的效果，显然是准确率为100%的情况最好。但是对于无监督算法，即聚类来看，标注效果显然不能达到令人满意的效果。而在各类视频网站上，要尽量避免将正常评论误判为垃圾评论的情况下尽量提高垃圾评论的识别率。

对于聚类结果的评价，需要根据具体任务需求选择合适的评价指标。对于垃圾评论检测任务，其目标是尽量减少误判率，因此需要以正常评论被错误分类为垃圾评论的数量为首要评价指标。同时，也需要保证垃圾评论被识别到的准确性，以保证标注的准确性和聚类结果的质量。因此，应该综合考虑正常评论和垃圾评论的识别率和准确性来评价聚类结果的好坏。

虽然聚类是无监督算法，但是由于任务特性所致，我们仍然利用了混淆矩阵对聚类效果进行评判，以判断通过哪些特征能够更加精确地识别出垃圾评论。

垃圾评论的消减是这个任务的关键目标，因此需要优先保证所有实际为垃圾评论的样本都能够被正确地识别和过滤掉，以最大程度地减少垃圾评论对正常评论的干扰。这时，召回率就显得尤为重要，它能够帮助评估分类器的分类效果，在一定程度上反映出已经发现的垃圾评论中被正确过滤的比例。

但是由于缺乏具体情境，因此聚类在测试集的成绩仅供参考。

### 聚类及结果展示

我们依次将CommentAuthorCounts、Polarity、CommentLength、WorthWordRatio、PCA、前四个特征、除CommentAuthorCounts和CommentLength之外的特征以及所有特征设置为变量进行八种聚类，并对每次聚类均设置迭代40次，以实现逐步寻优的目的。

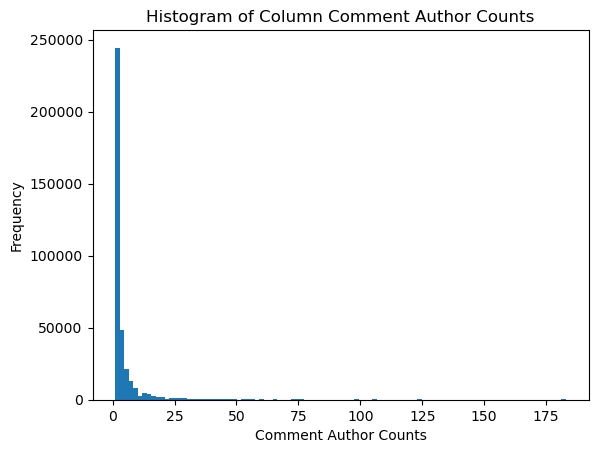
###### 3.1 对CommentAuthorCounts进行聚类

分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

从分布图中可以看出，此项有较为严重的偏态分布，且较多数据集中于10以内，即评论者多数仅评价少数次。

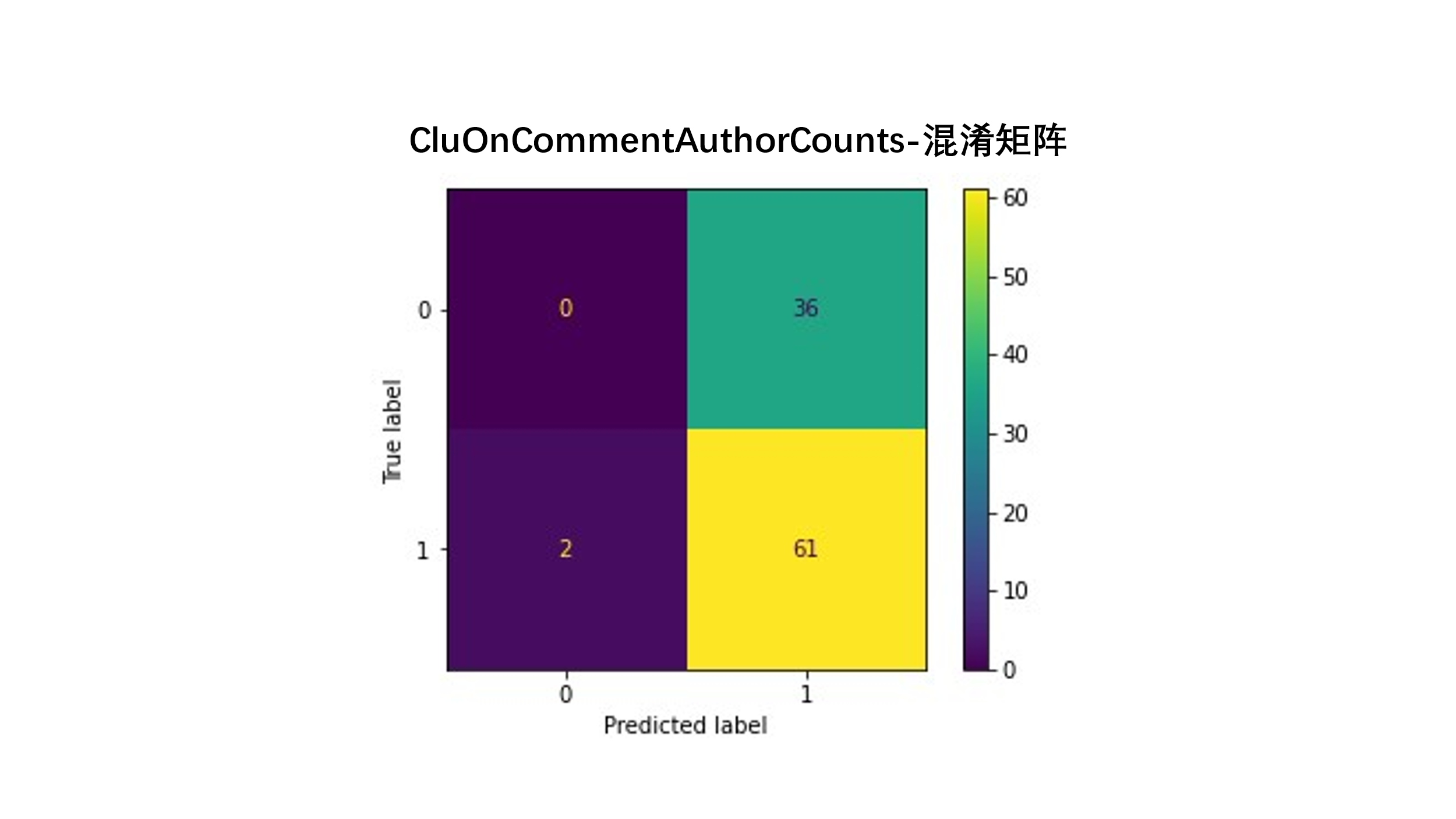
由于严重偏态分布，同时因为数据聚集在最小值，所以没有对数据进行对数处理，导致此处聚类效果并不好，具体表现为垃圾评论数目过少导致垃圾评论召回率过少。

因此，不能仅依靠此特征对垃圾评论进行标注。



|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| CommentAuthorCounts | 0.018799584 | 0.421068652 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 360150 |
|  | 2 | 6372 |
| 有效 | 366522 | |
| 缺失 | 0 | |



|  |  |
| --- | --- |
|  | CluOnCommentAuthorCounts |
| Recall\_score0 | 0 |
| Recall\_score1 | 0.9682539682539683 |

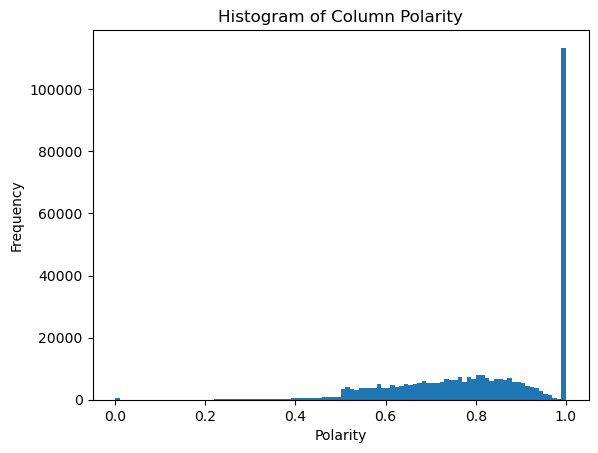
###### 3.2 对Polarity进行聚类

分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

从分布图中可以看出，排除情感极性为1的评论，数据具有较为规整的正态分布，原数据可看作一种较为极端的双峰分布，与我们小组预设的情形基本一致。

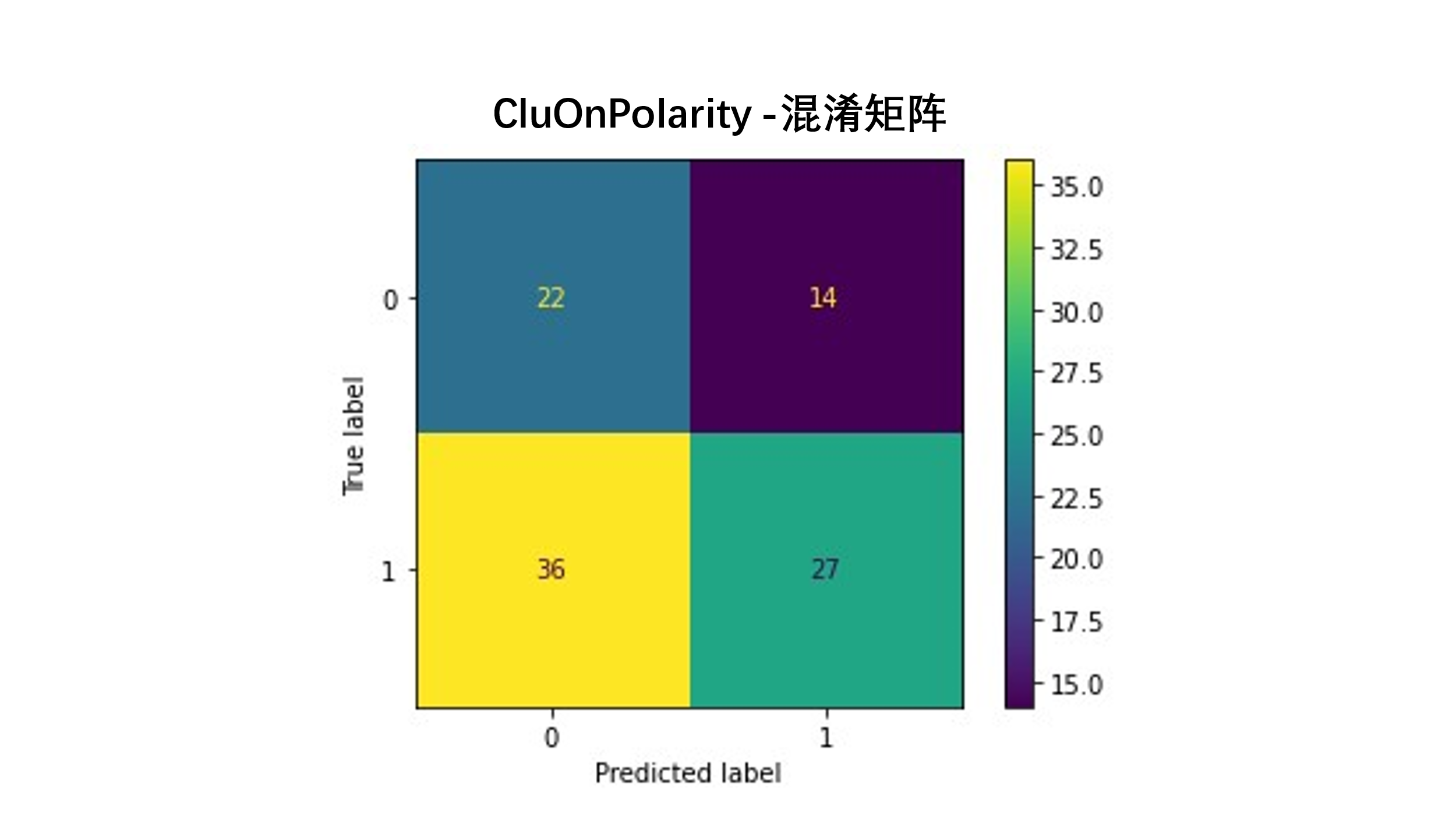
然而效果的表现并不是很好：准确率并不高的同时，正常评论的召回率过低。

因此，不能仅依靠情感极性对垃圾评论进行标注。



|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| Polarity | 0.641 | 0.936 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 153639 |
|  | 2 | 212883 |
| 有效 | 366522 | |
| 缺失 | 0 | |



|  |  |
| --- | --- |
|  | CluOnPolarity |
| Recall\_score0 | 0.611111111111111111 |
| Recall\_score1 | 0.42857142857142855 |

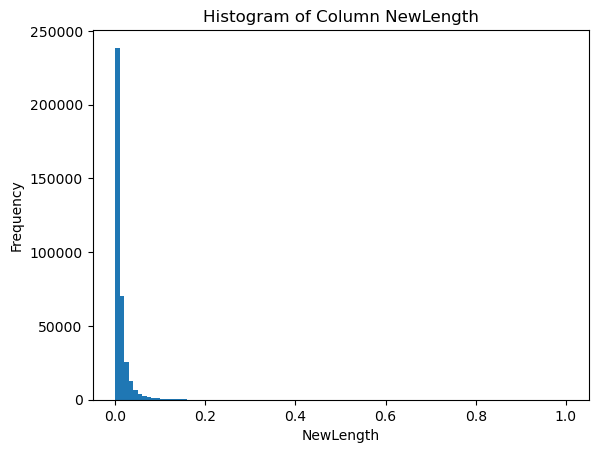
###### 3.3 对CommentLength进行聚类

分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

从分布图中可以看出，此项有较为严重的偏态分布，且较多数据集中于0.1以内，即评论多数属于短评。

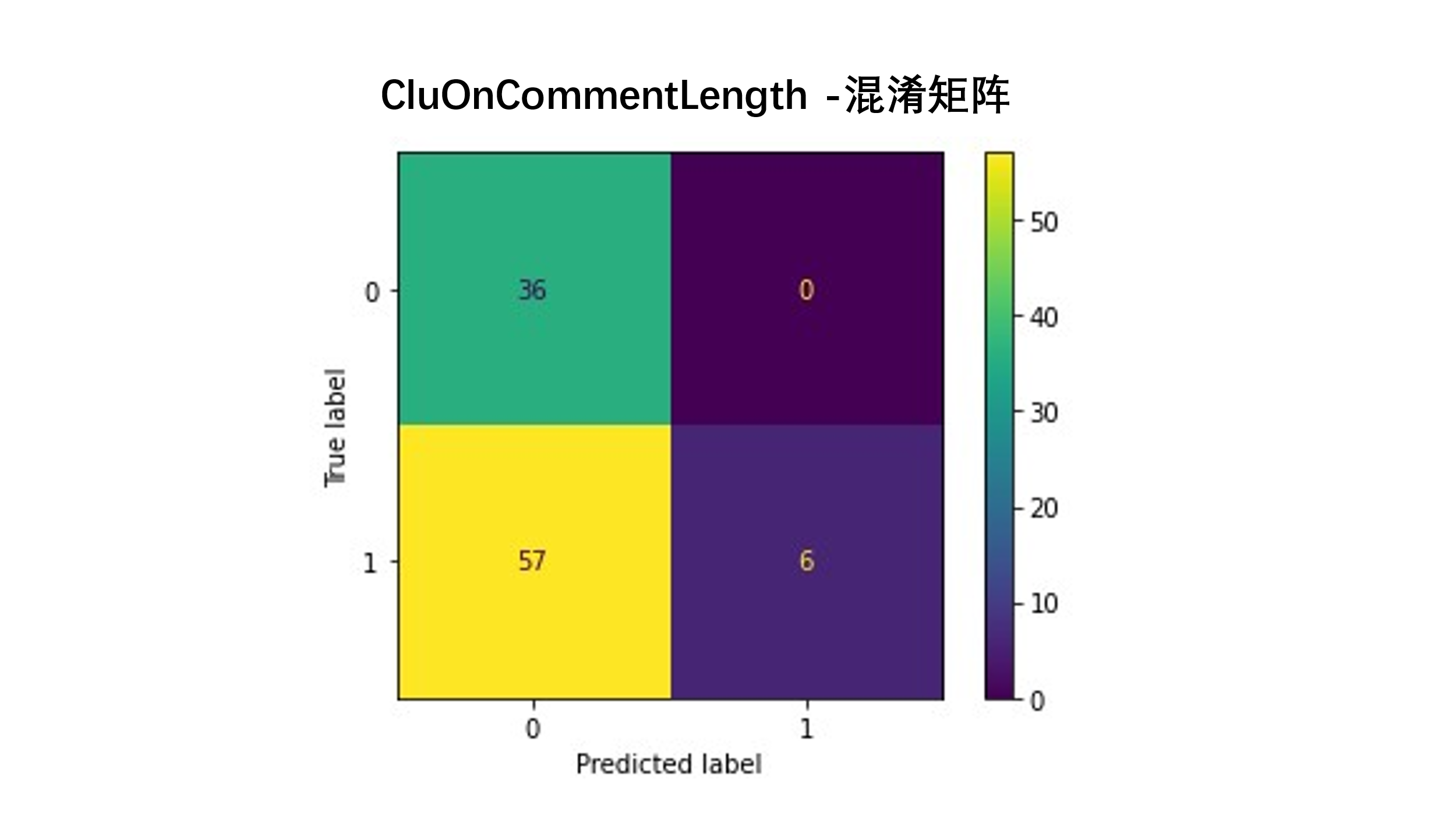
由于严重偏态分布，并且因为数据聚集在最小值，所以没有对数据进行对数处理，导致此处聚类效果并不好，具体表现为正常评论数目过少导致正常评论召回率过低。

因此，不能仅依靠此特征对垃圾评论进行标注。



|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| NewLength | 0.075822961 | 0.009219464 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 17858 |
|  | 2 | 348664 |
| 有效 | 366522 | |
| 缺失 | 0 | |



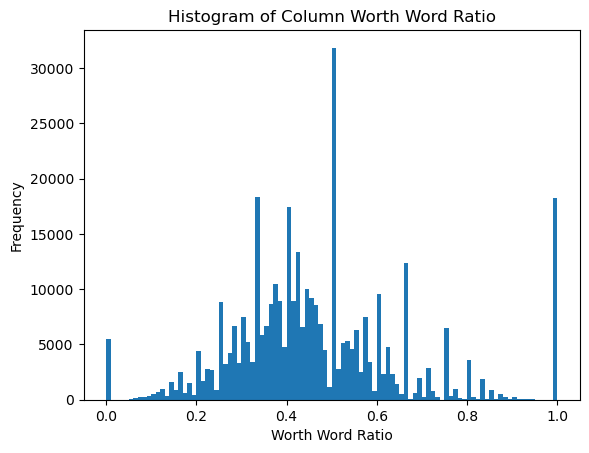
|  |  |
| --- | --- |
|  | CluOnCommentLength |
| Recall\_score0 | 1.0 |
| Recall\_score1 | 0.09523809523809523 |

###### 3.4 对WorthWordRatio进行聚类

分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

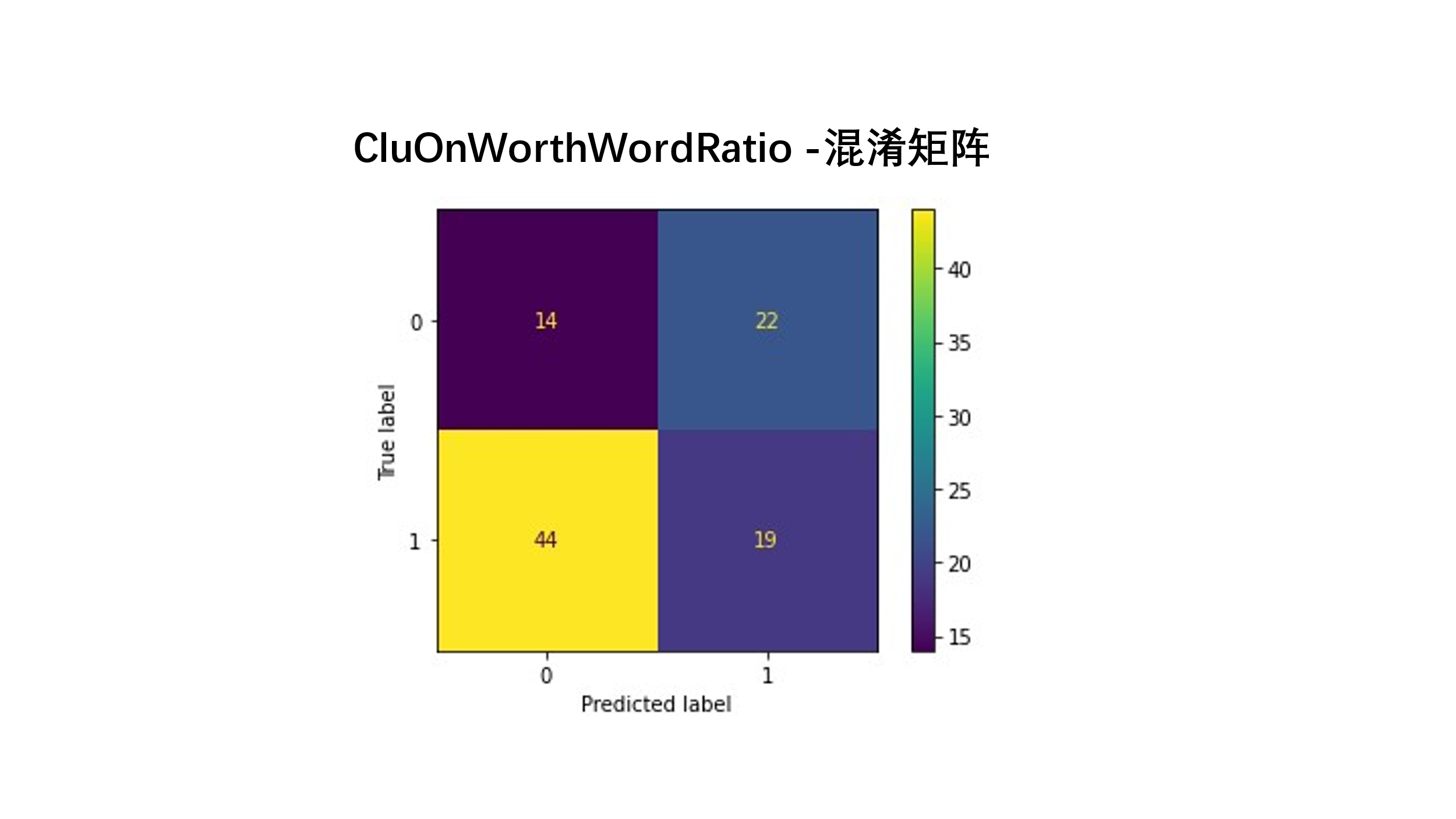
从分布图中可以看出，此项的分布大体上也可以说是呈双峰分布，数据主要分布在0.2到0.6之间。

聚类的结果标明以这一特征进行的聚类效果不佳，对于正常评论和垃圾评论的区分度较大。



|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| WorthWordRatio | 0.650 | 0.348 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 144649 |
|  | 2 | 221873 |
| 有效 |  | 366522 |
| 缺失 |  | 0 |



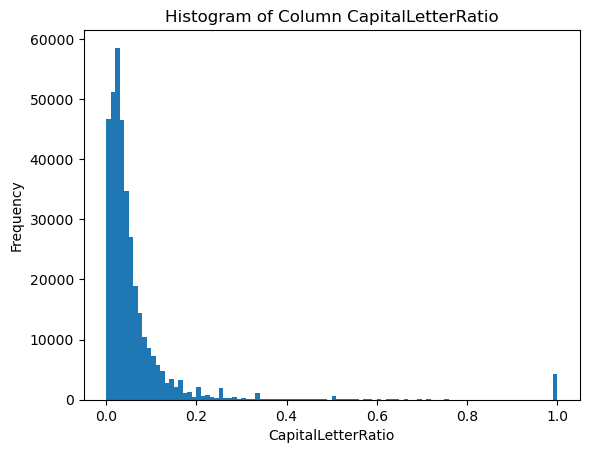
|  |  |
| --- | --- |
|  | CluOnWorthWordRatio |
| Recall\_score0 | 0.38888888888888889 |
| Recall\_score1 | 0.30158730158730157 |

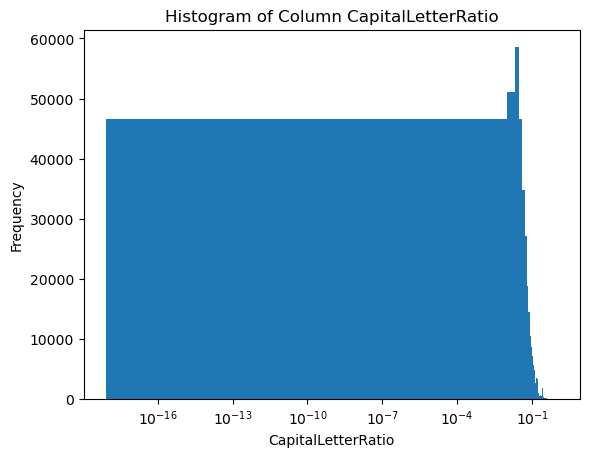
###### 3.5 对CapitalLetterRatio进行聚类

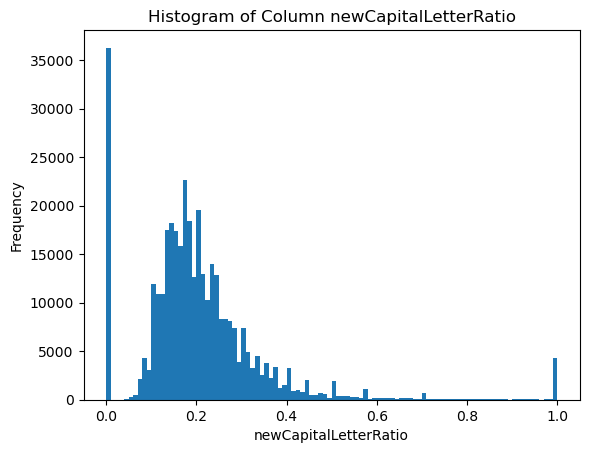
分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

从分布图中可以看出，此项有较为严重的偏态分布，并且在对数转换之后数据具有较为明显的均匀分布特征，因此最终对原数据采用方根处理，最终分布如newCapitcalLetterRatio所示。

根据这一特征进行的聚类倾向于认为评论是正常的评论，在测试集上识别正常评论的准确性达到了100%，但相对的，这一聚类识别垃圾评论的能力十分糟糕，几乎不能识别出垃圾评论，因此我们认为这一聚类的区分度是不够的。

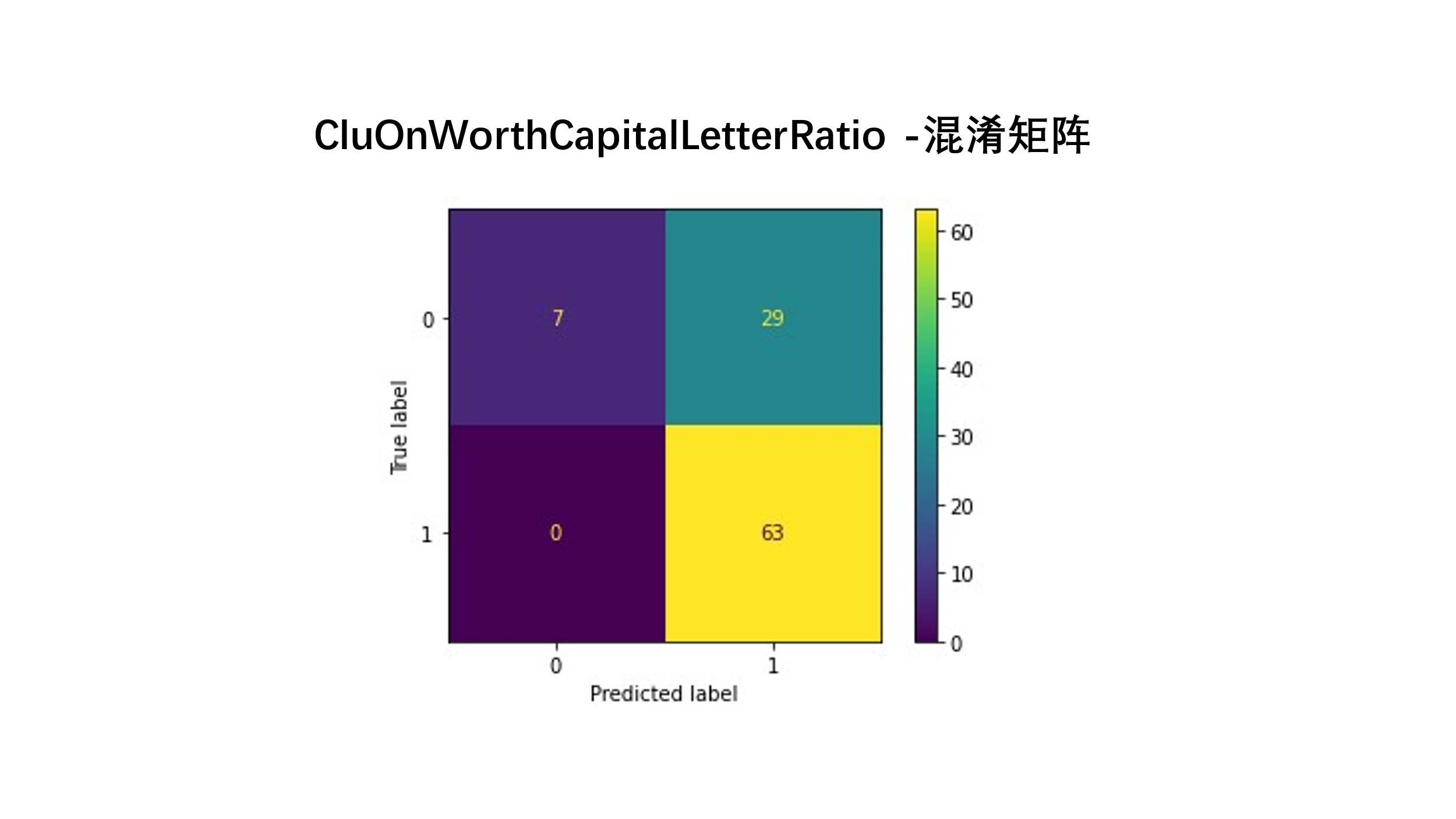






|  |  |  |
| --- | --- | --- |
|  | 聚类 | |
|  | 1 | 2 |
| WorthWordRatio | 0.182557768 | 0.636556727 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 346583 |
|  | 2 | 19939 |
| 有效 |  | 366522 |
| 缺失 |  | 0 |



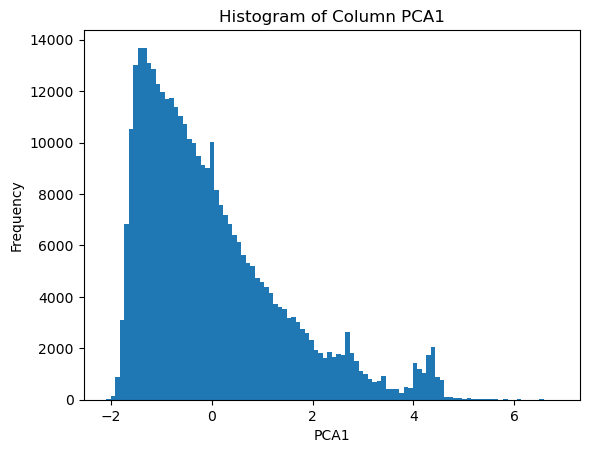
|  |  |
| --- | --- |
|  | CluOnCapitalLetterRatio |
| Recall\_score0 | 0.194444444444444 |
| Recall\_score1 | 1.0 |

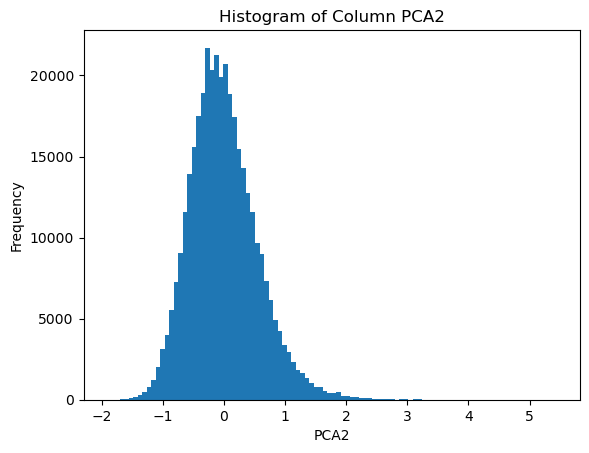
###### 3.6 对语义向量进行聚类

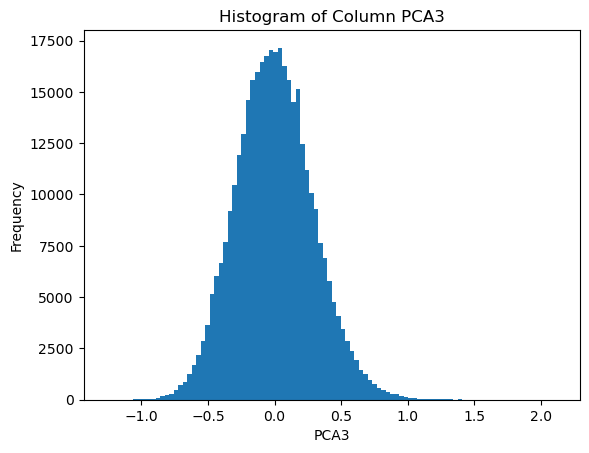
分布、最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

从分布图中可以看出，仅PCA1有明显的偏态分布，PCA2、PCA3都有理想的正态分布特征。然而从聚类效果图上来看，三个变量由于经过PCA处理，具有分布独立的特征，因此聚类效果并不理想。

此处聚类得出的轮廓系数为0.5273。



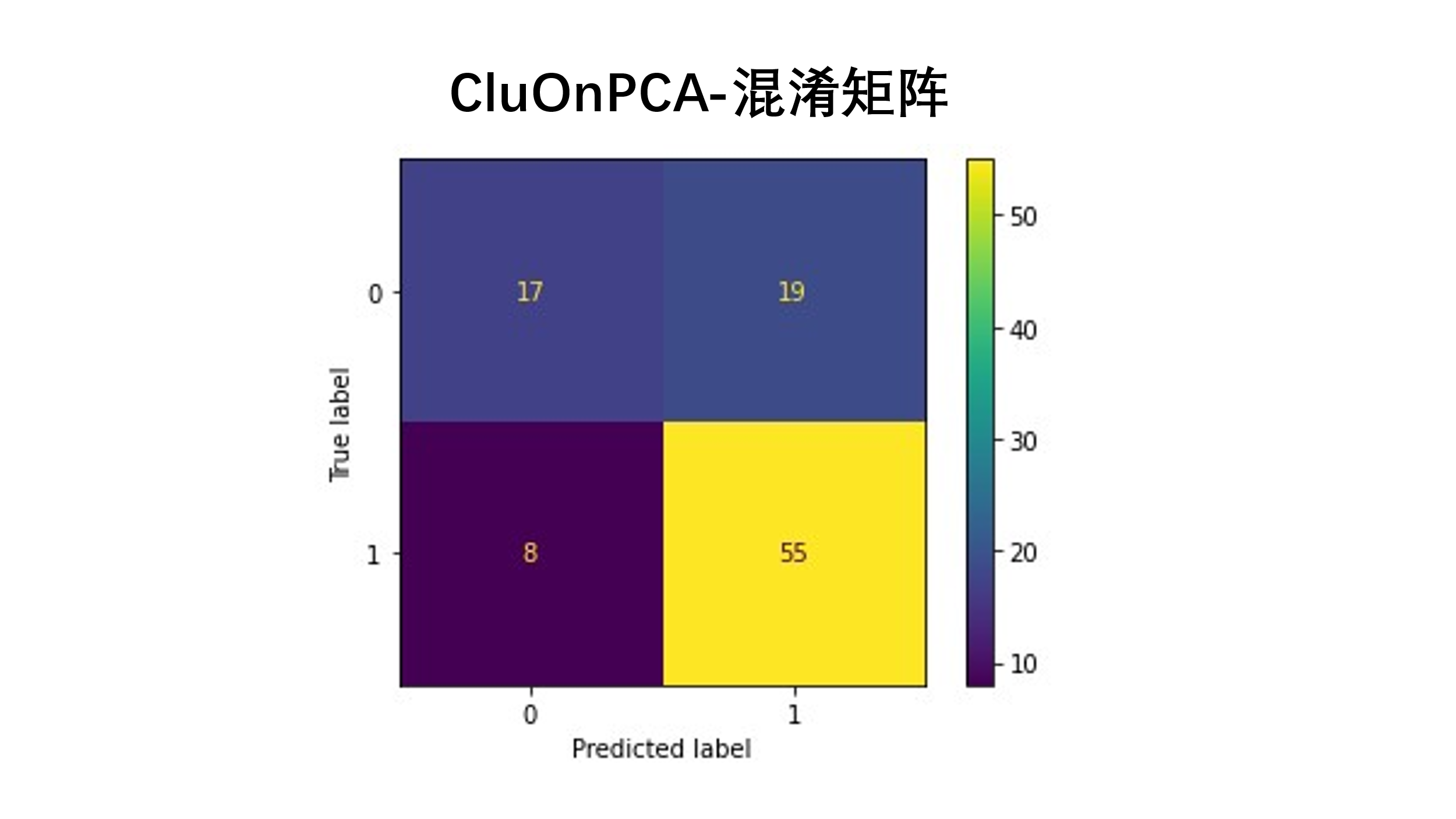




###### 

|  |  |  |
| --- | --- | --- |
|  | 聚类 |  |
|  | 1 | 2 |
| PCA1 | -0.69335689 | 1.96045797 |
| PCA2 | 0.00766488 | -0.02731381 |
| PCA3 | -0.00192753 | 0.00724801 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 271550 |
|  | 2 | 94972 |
| 有效 |  | 366522 |
| 缺失 |  | 0 |

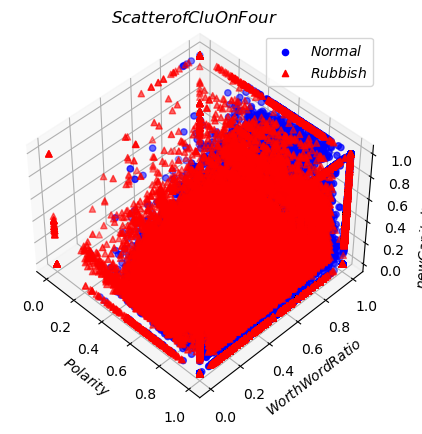


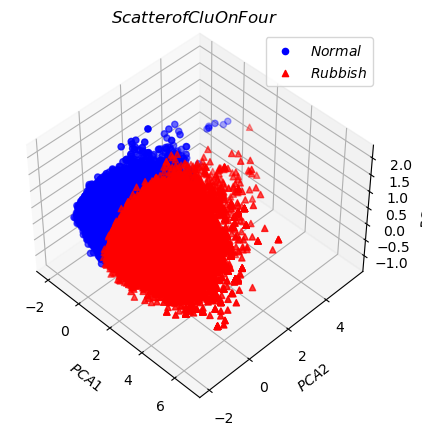
|  |  |
| --- | --- |
|  | CluOnPCA |
| Recall\_score0 | 0.472222222222222 |
| Recall\_score1 | 0.873015873015873 |

###### 3.6 对所选四个特征进行聚类

最终聚类中心、聚类个案数目、混淆矩阵与结果如下。

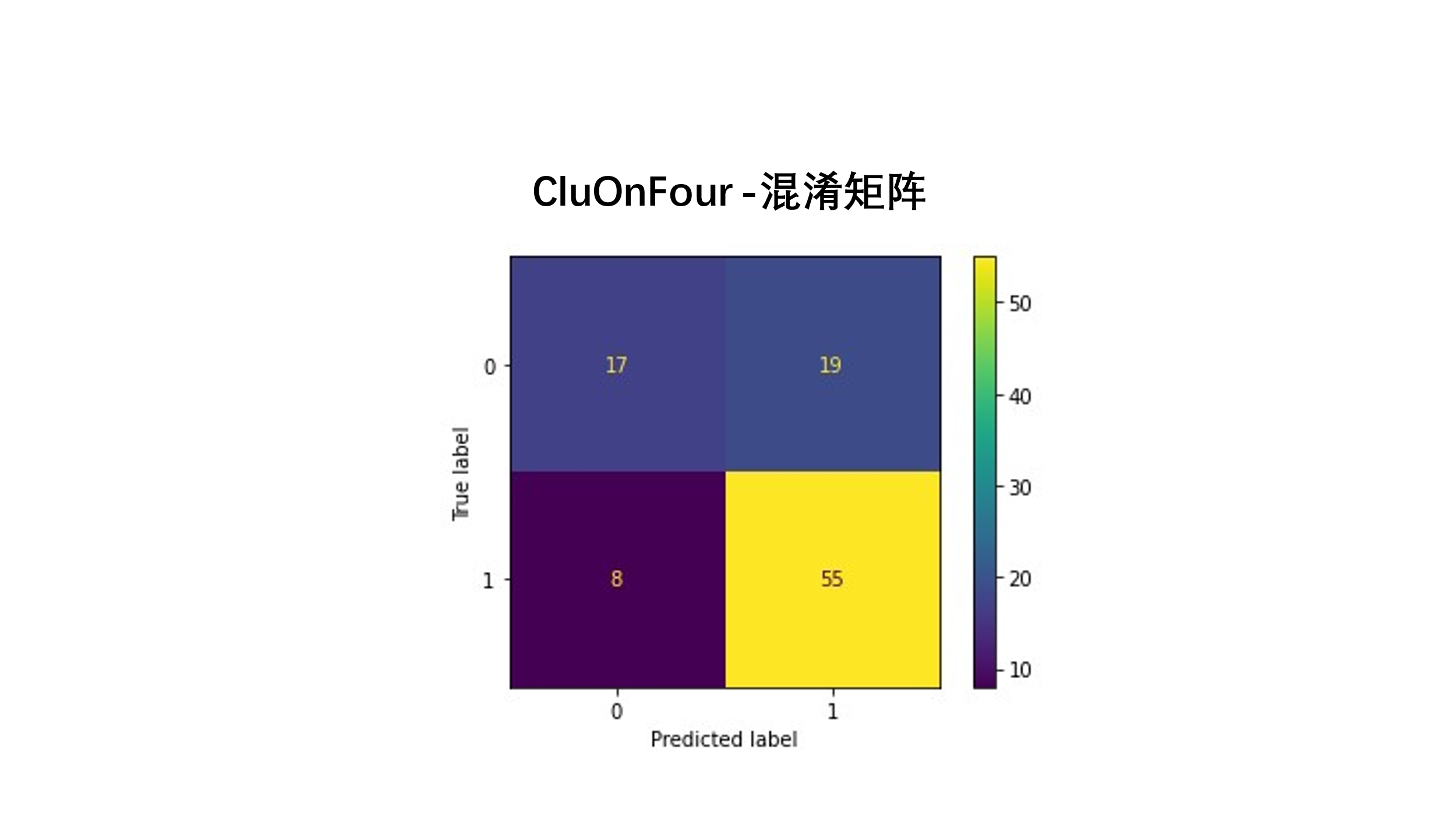
对四个特征进行的聚类实际上还是在对PCA进行聚类，由于这一部分的影响比较大，使得最终的结果的混淆矩阵表现的和PCA部分的混淆矩阵一致。





|  |  |  |
| --- | --- | --- |
|  | 聚类 |  |
|  | 1 | 2 |
| Polarity | 0.811 | 0.817 |
| WorthWordRatio | 0.443 | 0.537 |
| CapitalLetterRatio | 0.186262092 | 0.267254146 |
| PCA1 | -0.69334607 | 1.96045497 |
| PCA2 | 0.00760398 | -0.02714006 |
| PCA3 | -0.00203132 | 0.00754488 |

|  |  |  |
| --- | --- | --- |
| 个案数目 | | |
| 聚类 | 1 | 271518 |
|  | 2 | 95004 |
| 有效 |  | 366522 |
| 缺失 |  | 0 |



|  |  |
| --- | --- |
|  | CluOnThree |
| Recall\_score0 | 0.472222222222222 |
| Recall\_score1 | 0.873015873015873 |

## 总结

在该研究中，我们的目的是找出潜在的垃圾评论，对其进行删除并对相应的用户进行一定的处罚，同时要尽可能的少误伤正常的评论，因此，对垃圾评论的判断应该尽可能的保守，应该保证召回率尽可能的高。在本次选取的特征，对大写字母进行的聚类是召回率最高的，但很明显，采用这种方法会使得大量的垃圾评论也被认为是正常评论，效果非常差。综合来看，我们认为采用PCA进行的聚类是效果最好的，不过这种方法的一个主要缺点就是对垃圾评论的识别效果很一般，识别垃圾评论的准确性只在50%左右。

接下来讲讲我们模型存在的不足，我们认为主要的原因是选择的特征不够够，由于原数据集的缺陷，一些重要的指标，如用户过往发言记录和用户的好友网络等特征无法获取，因此本模型的预测准确率仍存在较大的改进空间，另一部分原因是因为k-means聚类对噪声和离群点敏感，降低识别的正确的，还有一部分原因是k-means聚类对簇形状和大小敏感，实际上，有用的评论往往是多于垃圾评论的（在正常的视频的评论区中）。

不过k-means本身在该领域也有其的优点，它的计算效率高，且不需要根据标注的信息来训练，因此我们认为这一方法比较适合初创公司前期用于识别平台上的垃圾评论。