深圳大学考试答题纸

(以论文、报告等形式考核专用)

二○ ～二○ 学年度第 学期

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  编号 |  |  | 课程  名称 |  |  | 主 讲  教师 |  | 评  分 |
| 学  号 |  |  | 姓  名 |  |  | 专业年级 |  |  |
|  |  |  |  |  |  |  |  |  |
| 教师评 | 语： | | | | | |  |  |
| 题  目： | **面向新闻文章的聚类算法** | | | | | |  |  |

# 引言

面向新闻文章的聚类算法是一种将新闻文章根据其主题或内容相似性进行分组的算法。在信息时代，人们可以轻松地获得大量的新闻文章，但如何对这些文章进行有效的分类和分析仍然是一个具有挑战性的问题。传统的分类方法通常需要人工标注数据并手动提取特征，工作量大且效率低下。在当前信息爆炸的时代，我们每天都可以获得大量的新闻信息，如何有效地对这些信息进行分类和整理已成为一个待解决的问题。传统的分类方法需要大量的人工标注和特征提取，效率低下且费时费力。因此，研究人员们开始探索使用机器学习和自然语言处理技术来解决这个问题。面向新闻文章的聚类算法正是在这样的背景下应运而生，它可以帮助我们快速、自动地对大量的新闻文章进行分类，从而更好地理解和分析这些文章。

研究面向新闻文章的聚类算法在新闻推荐、舆情监测和信息检索等领域都有着广泛的应用。在新闻推荐领域，面向新闻文章的聚类算法可以帮助我们将用户的兴趣和偏好与新闻主题相匹配，从而提高新闻推荐的精度和效果。在舆情监测领域，面向新闻文章的聚类算法可以帮助我们快速地发现和分析热点话题和舆情动向，从而更好地了解和应对公众的需求和反应。在信息检索领域，面向新闻文章的聚类算法可以帮助我们快速地找到与查询主题相关的新闻文章，从而提高信息检索的效率和准确性。

聚类算法是一种无监督学习算法，因为它不需要任何人工标注的信息就可以对数据进行分类。在新闻文章数据集中，我们可以使用聚类算法来将类似的文章归为一类，这样就可以更好地了解不同类型的新闻文章所反映的国家新闻格局等信息。

在本文中，我们将使用 K-means 算法对新闻文章数据集进行聚类分析。在使用 K-means 算法对新闻文章进行聚类分析时，我们需要先将文章转换为向量表示。常用的向量表示方法包括词袋模型、TF-IDF 模型和词向量模型等。具体而言，词袋模型将每篇文章表示为一个包含词频信息的向量，TF- IDF 模型将每篇文章表示为一个包含词频和逆文档频率信息的向量，而词向量模型则将每篇文章表示为一个低维稠密向量，反映文章中的语义信息。

在得到文章的向量表示后，我们可以使用 K-means 算法对文章进行聚类分析。首先，我们需要选择合适的簇数量K。一般来说，可以使用轮廓系数等指标来确定最佳的 K 值。接着，将文章向量输入 K-means 算法中，得到每个文章所属的簇。最后，根据聚类结果对新闻文章进行分析和可视化， 得出一定的结论。

# 相关原理

* 1. **词汇表的创建与简化**

在文本聚类中，词汇表的创建和简化是非常重要的步骤。词汇表是文本中出现的所有单词的集合。在文本聚类中，我们需要将文本中的单词转换为数字形式，以便计算相似度和距离。词汇表的创建包括以下步骤：

1、收集一个包含大量文本数据的数据集。

2、将文本数据集中的每个文本划分为单词序列，这个过程称为分词。在分词过程中，我们需要考虑如何处理停用词、数字、标点符号和特殊字符等。

3、将所有文本中出现的单词去重，得到所有可能的单词的集合。

4、将集合中的一些不必要的单词过滤掉，如停用词、低频词、高频词等。

5、对集合中的单词按照一定的规则进行排序，如按照单词出现的频率进行排序。

6、选择一定数量的单词作为词汇表。

由于互联网文本数据中常常包含非单词的字符和符号组合，如"@username"、"#hashtags"、"well..." 等。这些字符和符号组合不仅会增加数据集的维度，还会影响文本聚类的效果。不仅如此，文本还存在一些缩写、拼写错误和词形变化等问题，需要对其进行词形还原和词干提取，将单词转换为它

们的基本形式，以减小数据集的维度并提高文本聚类的效果。因此，在处理源自互联网的文本数据时，我们需要对词汇表进行简化，本文主要采用Python 中的NLTK 库对词汇表进行简化。

除此之外，如果直接将所有单词形成的词汇表用于后续的矢量化，会导致矢量化后产生更庞大的稀疏矩阵，增加计算和存储的复杂性。因此应当通过选取合理的分位数，以此简化词汇表。

# 词汇表的矢量化（TF-IDF）

算法的运行是对数值化的数据进行运算，在获得词汇表之后，需要将词汇表的文本内容转化为数据形式。词汇表的矢量化是一种常见的文本表示方法，它将文本数据转换为数值形式。

其中TF-IDF 是一种常用的矢量化方法。TF-IDF（Term Frequency-Inverse Document Frequency） 是一种用于衡量单词在文本中重要性的指标。它基于两个因素：单词在文本中出现的频率和单词在整个语料库中的重要性。TF-IDF 计算公式如下：

TF-IDF(w,d)= TF(w,d)\*IDF(w) (1)

TF(w, d)表示单词 w 在文档 d 中的频率，IDF(w)表示单词 w 的逆文档频率，定义为：

IDF(w)=log(N/(1+DF(w))) (2)

其中，N 表示语料库中文档的总数，DF(w)表示包含单词 w 的文档数。

TF-IDF 矢量化的过程如下：

1、计算词频：对于每个文档，计算每个单词在文档中的频率TF(w, d)。

2、计算逆文档频率：对于每个单词，计算它的逆文档频率 IDF(w)。

3、计算 TF-IDF 值：对于每个单词，在每个文档中计算它的TF-IDF 值，得到一个向量。

4、归一化：对每个向量进行归一化，以消除文档长度的影响。

TF-IDF 矢量化的结果是一个稀疏矩阵，其中每行表示一个文档，每列表示一个单词，每个元素表示该文档中该单词的TF-IDF 值。这个矩阵可以用于文本分类、信息检索等任务。

# PCA 降维

文本聚类中的 PCA 降维，通常是指将高维的文本特征表示（如词袋模型、TF-IDF 等）通过 PCA

转换成低维特征表示，以便于进行聚类分析。

PCA 降维的原理是通过线性变换将高维度的数据转换为低维度的数据，同时保留尽可能多的原始数据信息。

在文本聚类中，每个文本通常被表示为一个高维的向量，其中每个维度表示一个单词或短语在该文本中的出现次数或TF-IDF 值。这样的高维向量可能会导致聚类算法的计算复杂度非常高，同时也容易出现维度灾难等问题。因此，可以使用 PCA 算法将这些高维向量映射到一个低维空间中，从而减少维度并保留尽可能多的信息。在这个低维空间中，可以更加有效地进行聚类分析，同时也可以减少计算和存储的成本。

# K-means 聚类

K-means 是一种常用的聚类算法，它可以将数据集划分为 K 个不同的簇，每个簇包含一组相似的数据对象。其基本原理是将数据集划分为 K 个不同的簇，每个簇包含一组相似的数据对象。其主要步骤如下：

1、随机选择 K 个中心点作为初始簇中心：在数据集中随机选择 K 个数据点作为初始簇中心。可以使用随机选择或者其他选择方法来选取初始簇中心。

3、分配数据点到簇中心：对于每个数据点，计算它与每个簇中心的距离，将它分配到距离最近的簇中心所在的簇中。距离可以使用欧式距离、曼哈顿距离或其他距离度量方法。

4、更新簇中心：对于每个簇，重新计算它的中心点，即将该簇中所有数据点的坐标取平均值作为新的中心点。

5、重复步骤 2 和 3，直到满足停止条件。常见的停止条件包括簇中心不再发生变化、簇内平均距离小于阈值、达到预设的迭代次数等。

5 输出聚类结果：最终，K-means 算法将数据集划分为 K 个不同的簇，每个簇包含一组相似的数据对象。可以输出每个簇的簇中心点、簇大小和簇内的数据点等信息。

K-means 聚类的优点是简单易用，计算效率高，对于大规模数据集也有较好的表现。它也可以很好地解决一些经典的聚类问题，如球形簇和高斯混合模型。

# 基于 Python 的算法实现

## 算法流程



图 1 流程图

## 数据集简介与欠采样

该数据集是从archive.org 网站获取了某段时间的新闻文章，包含了 20 万篇左右，每篇新闻文章包含了 id、title、publication、author、date、year、month、url、content。但受限于电脑内存，本次实验将从中随机抽取 10 万篇作为实验数据。随机抽样后得到的部分数据可视化如图 2 所示。

图 2 部分数据可视化

## Python 代码实现

### 数据预处理

数据预处理在数据分析和机器学习任务中扮演着至关重要的角色，通过数据预处理，可以提高数据的质量和可靠性，减少不确定性，为后续的数据分析提供更可靠的基础。本次实验数据预处理的内容主要包括删除空值、去除文本中的数字和小数点符号、分词并还原英文单词原型和删除专有名词这几部分。

首先将content列存在空值的数据行删除。

data.dropna(subset=[**'content'**],inplace=True)

其次利用正则表达式将数字和标点符号删除，这是因为在某些情况下，数字和小数点通常不包含有意义的信息并可能对文本聚类的结果产生干扰或不必要的影响。因此可以被删除以简化文本表示。

pattern = **r'[0-9.]'**

text = re.sub(pattern, **''**, text)

然后利用英文分词算法对英文进行分词处理并还原英文单词的原型。分词是将文本划分为单词的过程，而还原则是将单词还原为其原始形式，去除词缀和复数形式等。这有助于在文本处理任务中减少词形变化带来的多样性，将不同形式的单词归一化为相同的基本形式，利于后续处理。

tokens = word\_tokenize(text)

stemmer = PorterStemmer()

tokens = [stemmer.stem(token) for token in tokens]

最后是进行词性标注并删除专有名词，如人名、地名等。其目的是为了排除一些专有名词对聚类产生干扰。

tagged\_tokens = pos\_tag(tokens)

tokens = [token for token, tag in tagged\_tokens if tag != **'NNP'**]

### 创建并简化词汇表。

词汇表可以统计所有文章中单词以及出现的次数，用于之后单词的矢量化。在处理文本数据时会面临非单词的挑战，因此需要剔除这些单词，只保留完整英语语料库中的单词。

english\_words = set(words.words())

filtered\_vocab = vocab[vocab[**'word'**].isin(english\_words)] filtered\_vocab = filtered\_vocab.drop\_duplicates(subset=[**'word'**])

选取合理的分位数，简化词汇表，过滤部分低频词汇，提取出更具有区分性和表达能力的特征。

### 词汇表矢量化

TF-IDF的计算公式：

TF-IDF(w,d)= TF(w,d)\*IDF(w) (1)

TF-IDF 是通过将 TF 和 IDF 相乘得到的结果，它可以同时考虑一个单词在给定文本中的频率和在整个文本集合中的重要性。在文本聚类任务中，通过计算每个文本中单词的 TF-IDF 值，将其作为特征向量表示文本。在代码实现时，使用了scikit-learn 库中的TfidfVectorizer 类，将文本数据集转换为 TF-IDF 特征向量矩阵。

vectorizer = TfidfVectorizer(vocabulary=text\_list) X = vectorizer.fit\_transform(data[**'content'**]) c=vectorizer.get\_feature\_names\_out()

tfidf\_matrix = pd.DataFrame(X.toarray(), columns=c)

### 降维

本次实验采用主成分分析（Principal Component Analysis，PCA）来降维。

当我们有一个高维数据集时，主成分分析（PCA）可以帮助我们找到一组新的低维表示，这些表示能够保留数据中的主要信息。下面是 PCA 步骤的详细说明：

PCA 的运算步骤如下：

1. 标准化数据：对原始数据进行标准化处理，使得每个特征具有零均值和单位方差。这是为了消除不同特征之间的量纲差异，确保每个特征对降维过程的贡献是公平的。
2. 计算协方差矩阵：基于标准化后的数据，计算特征之间的协方差矩阵。协方差矩阵的元素表示了对应特征之间的线性相关性。
3. 计算特征值和特征向量：对协方差矩阵进行特征值分解，得到特征值和对应的特征向量。特征值代表每个主成分所解释的方差，特征向量则是对应于特征值的主成分方向。
4. 选择主成分：根据特征值的大小，选择前 k 个特征值对应的特征向量作为主成分，其中 k

是降维后的维度。这些主成分按照特征值的降序排列，所以前 k 个主成分解释了总方差的大部分。

1. 数据投影：将原始数据通过主成分的线性组合进行投影，得到降维后的数据。新的低维表示是通过将原始数据与选择的主成分相乘得到的。

在 Python 实现降维操作时，可以使用 scikit-learn 库中的 PCA 类，对 TF-IDF 特征向量矩阵进行降维处理。

pca = PCA(n\_components=100)

tfidf\_reduced = pca.fit\_transform(tfidf\_matrix) tfidf\_reduced = pd.DataFrame(tfidf\_reduced)

### 聚类

K 均值聚类（K-means clustering）是一种常见且经典的无监督聚类算法，在应用 K 均值聚类之前，需要预先指定聚类的数量 K。选择 K 值对 K-means 聚类结果有重要影响，不同的 K 值可以导致不同的聚类结果。选择一个合适的 K 值可以更好地反映数据的内在结构。如果 K 值过小，可能会导致将不同的簇合并在一起，造成信息丢失。如果 K 值过大，可能会将同一个簇划分为多个小簇，导致聚类结果过于细分。因此需要结合轮廓系数等评价指标来选取合适的 K 值。

在此通过使用 scikit-learn 库中的 KMeans 类，对降维后的 TF-IDF 特征向量矩阵进行聚类，聚类的数量为输入变量 k 的值，并将聚类结果存储在labels 中。

kmeans = KMeans(n\_clusters=k, n\_init=**'auto'**) labels = kmeans.fit\_predict(tfidf\_reduced)

## 实验评价指标

本次实验的结果评价方法有两个，一是客观的评价指标（轮廓系数），二是主观的评价指标（主题词）。

1. 轮廓系数

轮廓系数（Silhouette Coefficient）是一种常用的聚类评估指标，用于衡量聚类结果的紧密度和分离度。轮廓系数的取值范围在[-1, 1]之间，越接近 1 表示聚类结果越好，越接近-1 表示聚类结果越差。

对于数据点 i，轮廓系数 s(i) 的计算公式为：

s(i) = (b(i) − a(i)) / max{a(i), b(i)} （4） 其中，a(i)表示数据点 i 与同簇内所有其他数据点之间的平均距离；b(i)表示数据点 i 与不同簇中

所有数据点之间的平均距离；max{a(i), b(i)}是 a(i)和 b(i)中的最大值。需要对每个数据点 i 计算轮廓系数，然后计算所有数据点的轮廓系数的平均值作为整个聚类结果的轮廓系数。

由公式可知，如果 s(i)接近于 1，表示数据点 i 与自己的簇内的其他数据点紧密相关，并且与其他簇的数据点相对较远，聚类结果较好。如果 s(i)接近于-1，表示数据点 i 更接近于相邻的其他簇， 而不是其所属簇，聚类结果较差。如果 s(i)接近于 0，表示数据点 i 与同簇内的其他数据点和其他簇的数据点距离相似，聚类结果可能存在重叠或边界模糊的情况。使用 scikit-learn 库中的metrics 模块， 计算了聚类结果的轮廓系数。

s = metrics.silhouette\_score(tfidf\_reduced, labels)

1. 主题词

TF-IDF 值反映了单词在文档集合中的重要性，具有较高的 TF-IDF 值的单词通常具有较高的区分度和代表性。在进行聚类后，可以计算每个簇中每个单词平均 TF-IDF 值，并选取每个簇中 TF- IDF 值较高的单词作为该簇的代表性单词。通过观察这些代表性单词，可以初步了解聚类结果并对每个簇进行判别。

根据聚类结果， 获取每个簇中 TF-IDF 权重较高的单词， 并将它们存储在一个字典

top\_words\_per\_cluster 中。

cluster\_indices = [i for i, label in enumerate(cluster\_labels) if label == cluster] cluster\_docs\_tfidf = tfidf\_matrix.iloc[cluster\_indices]

avg\_tfidf = np.mean(cluster\_docs\_tfidf, axis=0) top\_word\_indices = avg\_tfidf.argsort()[-10:][::-1] top\_words = [word\_list[i] for i in top\_word\_indices] top\_words\_per\_cluster[cluster] = top\_words

# 实验结果分析

## K 值的选择

轮廓系数（Silhouette Coefficient）是一种用于评估聚类质量的指标，它结合了内聚度（Cohesion） 和分离度（Separation），可以用来衡量聚类结果的紧密度和分离度。

计算不同的 K 值对应的轮廓系数，选择轮廓系数最大的 K 值作为最优的 K 值。这种方法的优点是不受聚类方法和距离度量方法的限制，可以应用于各种聚类算法和数据集，但计算轮廓系 数可能比较耗时。

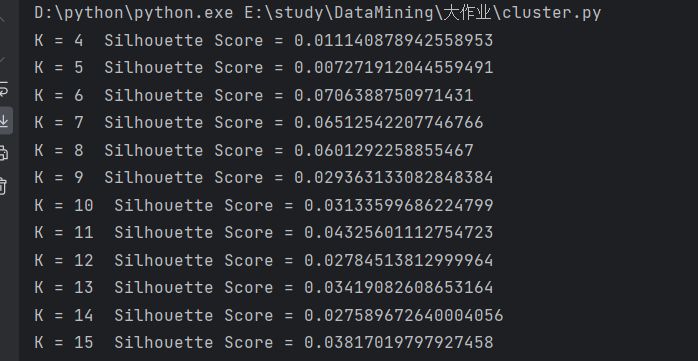
在本实验中为了寻找一个比较合适的 K 值，在词汇表分位数选择40%到99.5%，PCA降维维度选择100时，输出了K 从 4 到 15 的轮廓系数（图 3）。从输出结果可以看到，在这个聚类算法下，聚类结果的轮廓系数一般在 0 到 0.1 之间。

图 3 不同K 值下的轮廓系数

为了进一步选择 K 的具体中，在不同 K 值下算法分别运行 5 次并计算轮廓系数的 5 次平均

值，整理得到以下折线图（图 4）。由于当 K 取 6 时的聚类结果普遍好于其他 K 值，因此选择使用

K=6来进行聚类实验。

轮廓系数平均值

0.0800

0.0700

0.0600

0.0500

0.0400

0.0300

0.0200

0.0100

0.0000

0.0682

0.0508

0.0542

0.0365

0.0333

0.0368

0.03210.03330.032

8

0.0240

0.02840.0276

4

5

6

7

8

9 10 11 12 13 14 15

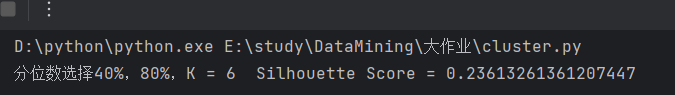
K值

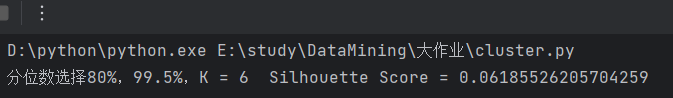
图 4 取不同K 值聚类多次的轮廓系数平均值

## 分位数选择

在构建词汇表时，需要选择一个合适的词汇表大小，通常可以通过设置词汇表中单词出现频率的分位数来控制词汇表的大小。

具体来说，分位数的选择可以根据实际情况进行调整，一般来说，分位数越小，保留在词汇表中的单词数量就越多，词汇表也就越大；分位数越大，保留在词汇表中的单词数量就越少，词汇表也就越小。

实验分别选择词汇表中分位数 40%到 80%，80%到 99.5%，40%到 99.5%的词汇进行了聚类，其中 K 的取值为 6。结果如图 3 所示，当选择词汇表 count 中第 40 分位数和第 80 分位数之间的词汇时，轮廓系数比较高，而选择第 80 分位数和第 99.5 分位数之间的结果和选择第 40 分位数和第 99.5 分位数之间的结果轮廓系数相近。



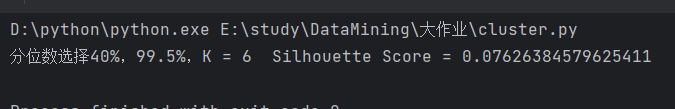


图 5 选择不同分位数聚类的轮廓系数

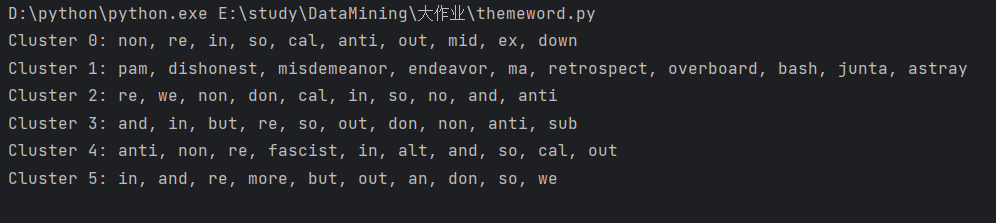
为了更为合理的对比不同的实验结果，选择最合适的分位数。将聚类结果中每一簇文本的关键词输出。结果如下 ：

图 6 分位数 40%,80%

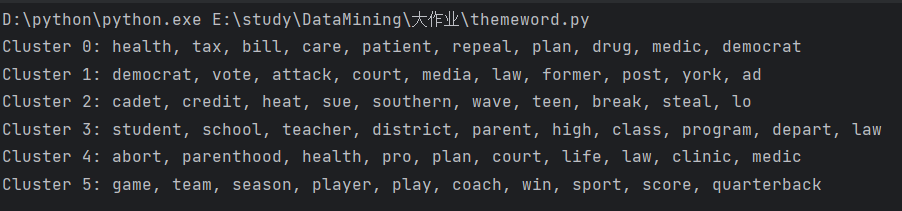


图 7 分位数 80%，99.5%

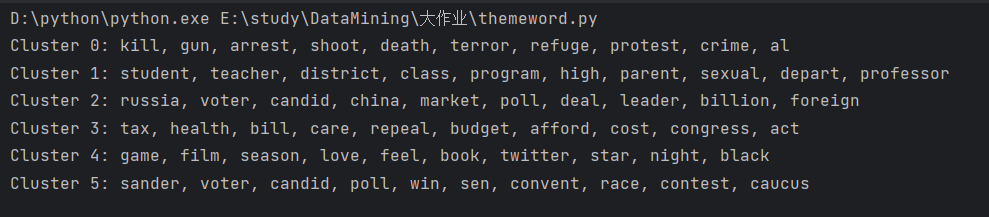


图 8 分位数 40%，99.5%

通过图 4 可以发现，当选择词汇表 count 中第 40 分位数和第 80 分位数之间的词汇做聚类时， 虽然轮廓系数表明聚类的效果较好，但是每一簇文本中的关键词（即 TF-IDF 值最高的词语）多为无意义的单词。当选择第 80 分位数和第 99.5 分位数之间的词汇做聚类时（图 5），在每一簇文本中提取的关键词能够较好的反映此类文本的主题。因此可以认为大部分比较重要的单词都会存在于词汇表频率的后 20%中。

但是如果仅仅选择后 20%的词汇进行聚类可能会忽略一些出现次数较少但重要的单词，这些单词可能对文本的主题和内容具有重要的贡献，忽略它们可能会导致聚类结果的信息损失。选择第 40 分位数和第 99.5 分位数之间的词汇做聚类分析（图 6）不仅容易区分每一簇的主题而且更多地保留了文本的信息。

因此选择词汇表 count 中第 40 分位数和第 99.5分位数之间的词汇进行实验并分析结果。

## 4.3 聚类结果分析（K=6，分位数 40%、99.5%，维数=100）

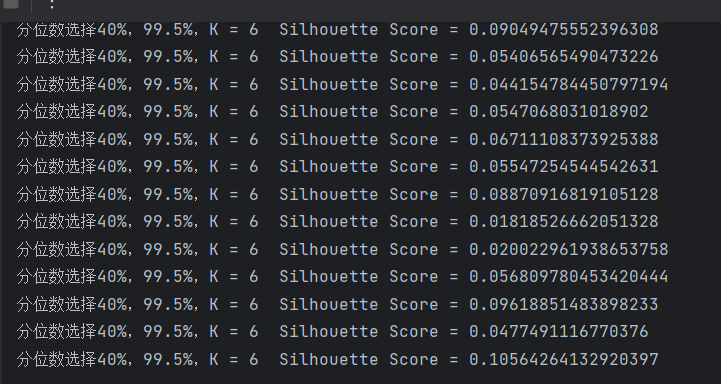
为了得到一个较好的结果，多次运行聚类算法得到一个较高的轮廓系数后分析该结果（图 9），在聚类结果分析时，先输出每一聚类簇中的关键词，初步判断每一类的文本主题（图 10）：

图 9 多次运行聚类算法

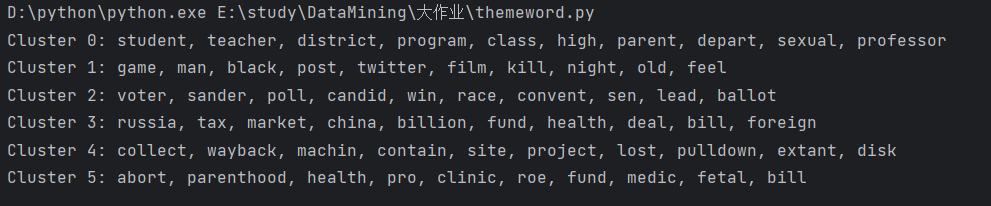


图 10 每一簇中的关键词

总结所输出的关键词可以大致总结这 6 簇新闻的主题：簇 0：Education（教育）；簇 1：Entertainment and Social Culture（娱乐与社会文化）；簇 2：Politics（政治）；簇 3：International Economics and Policy

（国际经济与政策）；簇 4：Data collection and storage technology（数据收集与储存技术）；簇 5： Reproductive health and legislation（生殖健康和立法）。

进一步统计每个簇中文章的百分比（图 11）：

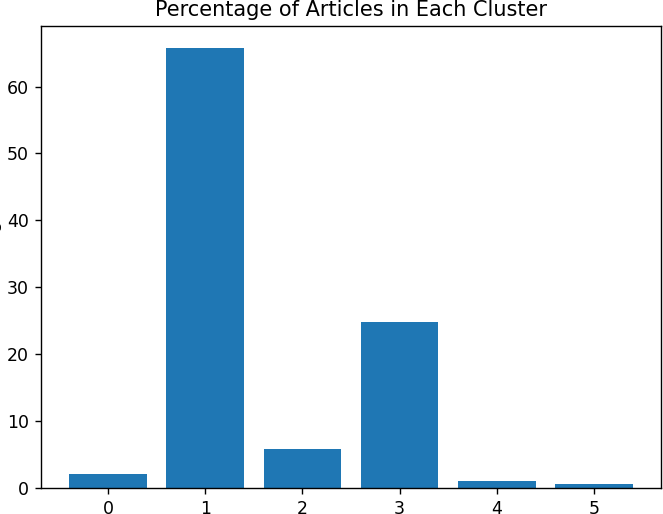


图 11 每个簇中文章的百分比

显然，在这些新闻数据集中，大多数的新闻都与社交娱乐有关，同时政治新闻和国际经济新闻也占有一定的比重，而关于数据储存技术和生殖健康方面的文章则比较少。

这个结果说明了娱乐与文化是大众关注的焦点，各大媒体对此的报道肯定会比较多。政治新闻和国际经济新闻是全球关注的重点话题，也是各大媒体报道的重点之一。至于数据储存技术和生殖健康方面的文章比较少，可能是因为这些领域相对比较专业和细分，且不太容易引起大众的广泛兴趣。

为了进一步分析聚类结果，分别输出每一簇中关于 publication 的统计关系：

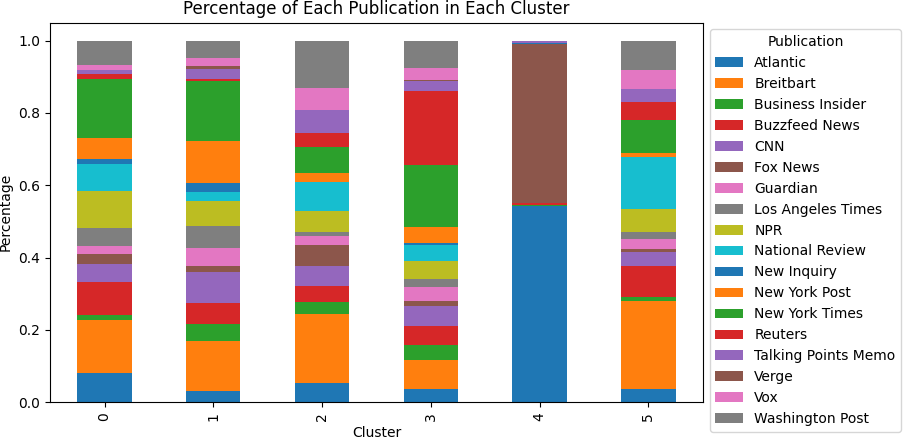


图 12 每一簇中每个出版社的占比

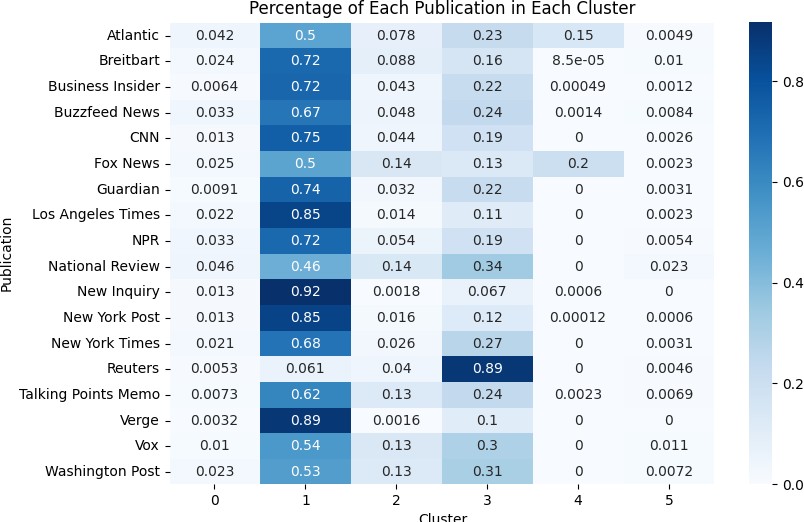


图 13 每个出版社在每一簇中的占比

结合图 12、13 可以得到结果，大多数出版社的在社交娱乐角度上有着大量的新闻报道，但是也显示出了部分新闻机构的自身特点：

* + 1. Reuters（路透社）着力于报道国际经济与政治类新闻，是一家国际性的新闻机构。
    2. New Inquiry 九成的报道都被分在娱乐与社会文化簇中，该机构的新闻报道主要侧重在社会热点事件，对于政治、经济、科技等方面涉及较少。

输出每一簇中关于 month 和 year 的统计关系：

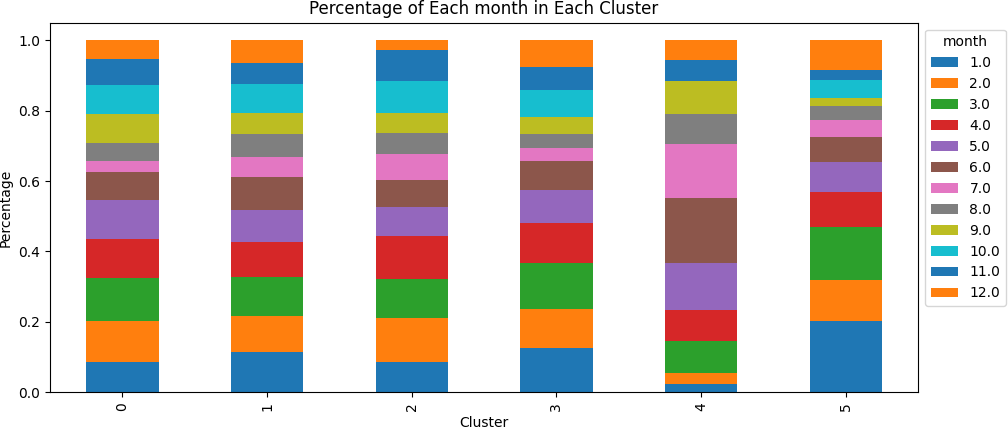


图 14 每一簇中每个月份的占比

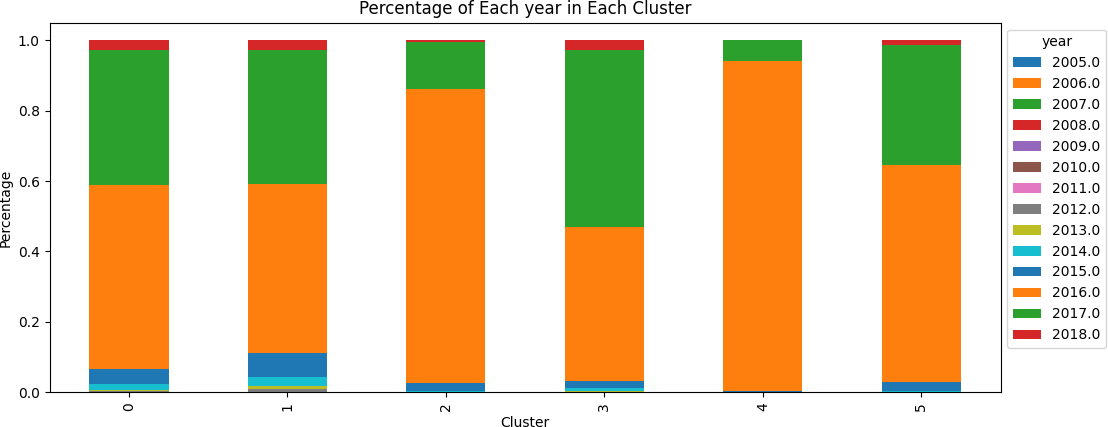


图 15 每一簇中每个年份的占比

根据图 14 和图 15 能够得出一些结论：

（1） 簇 4 即关于数据收集与储存技术的相关新闻报道大部分集中在 2016 年的 3 月至 9 月，推测在这一段时间里发生了与数据技术相关的社会事件。

（2）

# 总结