2021 年西北大学
数学建模协会赛

题目：C 题（ 填写 A、B、C 题 ）

	参赛队员 1	参赛队员 2	参赛队员 3
姓名	樊泽瑞	郭明皓	刘宇哲
学号	2020111089	2020111109	2021117188
学院	信息科学与技术学院	化学与材料科学学院	信息科学与技术学院
专业	软件工程	化学类	电子信息类
电话	15389422621	18392492712	13359206778
Email	477706421@qq.com	3053351813@qq.com	2309331399@qq.com

西北大学数学建模协会

影视评价与制定的优化模型建立

摘要

影视评价与定制问题基于大数据所提出，该问题受到水军刷票数据造假，夸大宣传等因素影响。为了解决这些问题，从互联网上搜索可靠的数据，过滤掉有瑕疵的值，确定权重并给出合理的预测是非常重要的。

针对问题一。在电视剧排名方面，我们选择电视剧得分、每部电视剧的评论人数和电视剧集数作为前三个重要指标来判断最终的排名。为了找到这三个指标中最合理的权重，我们使用 AHP 层次分析法来计算最佳权重，提供了一个新开发的指数 R 分数。在计算最终结果时，分别定义了三个加权值。

针对问题二，我们认为，判断明星流行度的指标是多样的，并不存在唯一客观指标。因此我们选择使用 Apriori 算法来过滤不重要的指标，只保留高权重指标。通过遍历从互联网收集的数据集，我们得到最终的频繁 n 项集作为其中最重要的指标。然后利用主成分分析法确定相关指标的权重。除此之外，我们还应考虑特殊情况，如花絮在短时间内引起的剧烈变化，二创作品的传播，明星个人的生日纪念日。最后，我们将基于我们的指数的排名与官方网站中存在的排名进行比较，发现近似相等。

针对问题三，为了建立一个新的团队来创造新产品，我们可以使用爬虫工具在互联网上获取公开数据，如点击率、评论、明星、制作团队等。为了过滤掉不重要的指标，我们采用逐步回归的方法，然后经过标准化得到回归方程。通过这个等式，每个指数将对应一个权重，该权重衡量对最终指数的贡献。然后，将导出的排名与官方排名进行比较，以获得可信度，并判断该指数是可接受的。根据最终指标，描述一个理想的影视剧团队。

针对问题四，为了从观众的浏览历史和每个频道的评分中获得最合适的推荐。这里我们选择使用 PUM-CF 算法生成推荐模型，CF 算法的汇总的是所有的<user,item>行为对，可以通过已有的数据集训练，推出针对个人用户的个性化推荐模型。我们将把推荐算法和聚类操作相结合，提出了一种高效的推荐算法，并使用经典的数据集进行了验证分析。

每一个模型所使用的数据确保真实，数据来自互联网，由 Python 中的 selenium 库配合 BeautifulSoup 库进行数据收集

1. 问题重述

1.1 问题背景

目前中国电视市场规模大，竞争激烈，产品类型多。每年虽然可以产出大量电视剧，但过多的电视剧缺少电视台购买，造成了大量的投资浪费。其核心原因在于大量电视剧质量差。为了提升电视剧质量，得到更大的利润回报，需要对于如何评价和定制影视剧等问题进行预测很分析。

已知大数据作为分析工具可以非常精确地分析数据和预测。这可以应用于剧本写作，电视评级预测，电视广告的结果和电视剧购买。可以降低电视投资风险，提高脚本质量，并预测受众响应以确保最大的收益。

1.2 问题重述

现在给出两个附件，其中附件 1 包含了 429 部电视剧的电视剧评分，电视剧评论数和电视剧所属类型的统计结果。数据为空说明缺少相关数据。其中附件 2 包含 429 部电视剧制作公司，发行时间等基本信息。

问题一：对于附件一 429 部电视剧的电视剧得分、每部电视剧的评论人数和电视剧集数进行分析评价，从电视剧自身质量出发，选出最优的十部电视剧。

问题二：通过信息搜集一年内各平台明星的相关热度信息，对其进行分析评价，建立模型设定明星人气指数显示明星个人的真实人气，同时今年的实例证明该模型的可靠性

问题三：根据目前已有的明星热度数据和观众画像，推测观众感兴趣的部分，由此描述一个理想的制作团队阵容。

问题四：建立根据观众个人的历史观看记录，为其定制推广其最感兴趣，最适合该观众的相关的节目的模型方案

2. 问题分析

2.1 问题一分析

我们首先通过主观判断，可以明确在论坛中所给出的评分高并不代表着电视剧一定是一部好剧，例如说：A 剧和 B 剧同样是 4.8 分，但是由于这两者的评论数不同导致这两部电视剧无法进行比较。大部分观众只是在电视剧播出之前提前通过观看评论判断出它是一部好剧，评论的走向也会很大程度的影响这部电视剧的观看意向。由于评论分数与评论数量之间存在偏差，我们需要一种综合其他从属因素的方法来完善电视剧的评价标准。经过认真的讨论，除前一个主要参数外，还考虑了剧集和著名明星的数量，以构成一个数学模型。

2.2 问题二分析

对于明星在大众中的受欢迎程度，我们选择使用 Apriori 算法进行处理。我们使用 32 个可能相关的条件，通过对其**支持度的阈值**的计算来进行合适的项集选择。

对于支持度的阈值和置信度的计算公式如下所示：

$$\text{Support}(X,Y) = P(XY) = \frac{\text{number}(XY)}{\text{num}(\text{AllSamples})}\text{（支持度）}$$

$$\text{Confidence}(X \Leftarrow Y) = P(X|Y) = P(XY)/P(Y)\text{（置信度）}$$

通过筛选，我们最终得到了 14 个相关的流行的评价指标。

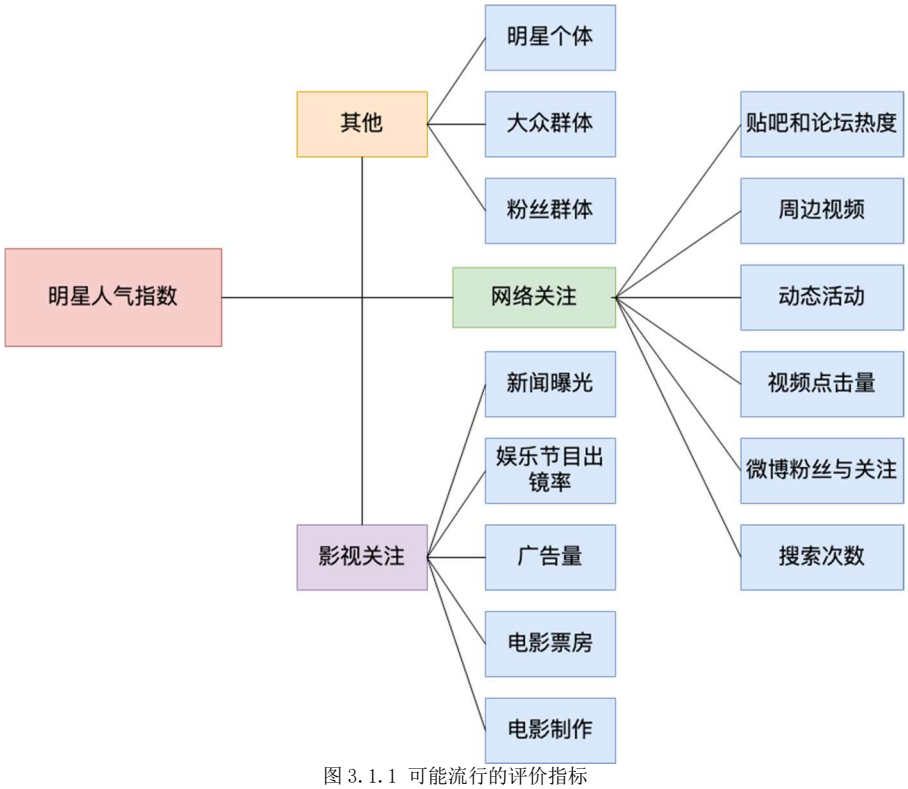


图 3.1.1 可能流行的评价指标

上述评价指标的受欢迎程度，虽然部分受欢迎指数对受欢迎指数贡献显著，但会随着其他指标的重复，其重要性被消减。

因此，我们可以利用**主成分分析法**对指标进行分析，得到一些贡献较大的指标，而最终的指标没有或几乎没有一致性。

2.3 问题三分析

对于市面上所能够见到的各式各样种类的电视剧，都有着不同的观众受众，为了找到不同观众对于何种电视剧更加感兴趣，我们首先收集了 2021 年年度人气电视剧排行榜：

排行	电视剧名称	参演明星	电视剧类型	电视剧集数
1	谁是凶手	赵丽颖,肖央,董子健,姚安濂	悬疑， 犯罪	16
2	风起洛阳	黄轩,王一博,宋茜,宋轶,咏梅	剧情,悬疑,古装	39
3	就这样…	莎拉·杰茜卡·帕克,克里斯汀·戴维斯,辛西娅·尼克松	剧情,喜剧,爱情	10
4	爱很美味	李纯,张含韵,王菊,周澄奥,刘冬沁	剧情,喜剧,爱情	20
5	女心理师	杨紫,井柏然,王嘉,菅纫姿,黄觉	剧情	40
6	那年，我们的夏天	崔宇植,金多美,金圣喆,卢正义	爱情	16
7	华灯初上	林心如,杨谨华,杨祐宁,凤小岳,张轩睿	剧情,爱情,悬疑,犯罪	8

表 4 2021 年年度人气电视剧排行榜

观众的兴趣往往是导演们所需要考察的对象。即使电视剧的导演如何去宣传自己所拍摄的电视剧，但是如果观众对于这部电视剧表示不买账。因此，以下想法旨在调查观众的兴趣。根据表 4 我们可以看到，于 2021 年喜剧和爱情剧的观众受众十分的广泛。

观众的评论往往反映了一部电视剧的实际情况，根据我们的常识而言，意见往往会在某种程度上影响某些人的观影体验，特别是我们面对一些资深观影人的评论时。因此我们提出使用逐步分析法对此问题进行求解。

2.4 问题四分析

在大数据信息爆发的时代，智能推荐作为一种能够针对于使用用户的个性化特征推荐相关内容的方式，面对数量如此庞大的影片资源和海量的用户群，如何对不同的用户进行精准、个性化的推荐，成为各大平台网站或手机 APP 提升自身竞争力的关键。而正因为影片资源和用户的数目都如此庞大，单纯依靠人力来实现个性化推荐已不具可行性，智能推荐系统以其高效、精准和即时的大数据处理能力而被各大平台网站或手机 APP 广泛采用。借助于注册信息、位置识别、个人通信录等来获得用户的性别、年龄、区域位置、文化程度、经济收入、社交人脉等个人信息，在此基础上进行“用户画像”，预测用户的兴趣或需求，建立用户和电影资源之间的匹配关系，并使每一个用户获得与其他用户不同的“精准”、个性化推荐，做到不同用户界面的“千人千面”，从而提升用户对平台的满意度和黏合度。

因此准确的根据用户的社会属性、个人行为、偏好兴趣等信息凝练出用户标签成为了至关重要的行为。用户画像的建立离不开大量的数据支撑，大数据时代一切数据都变大可视化，我们通过对于用户历史浏览的信息还有针对于电视台播放量等能够获取到的信息进行处理，得出一个合理的推荐模型，从而对于用户习惯向他进行个性化、精准的推荐视频，提高平台的收视率以及网站流量。

3. 模型假设

1）数据可以正确反映受欢迎程度，没有互联网恶意灌水等非自然趋势提高受欢迎程度的行为。
2）所有以明星或电视剧名称命名的论坛都在谈论相关主题。
3）对知名网站的排名没有商业猜测，所有排名都依赖于真实数据，并且必须是客观的。
4）这些模型具有普适性，因为来自互联网的数据不能包含所有明星和电视剧，通过计算足够大的数据规模来考虑。派生的模型可以适用于所有明星和电视剧，并且产生的误差应该足够小。

4. 符号说明

符号	说明
RScore	排名指数
S	评价分数
Chs	归一化评论数
Ehs	归一化剧集数
P_1	明星人气指数
P_2	影视人气指数
R	User-items 特征矩阵

表 1 符号说明

注：其他符号将会在文章中给出

5. 模型的建立与求解

5.1 问题一模型建立与求解

5.1.1 模型提出（AHP 层次分析法）

层次分析法是指将一个复杂的多目标决策问题作为一个系统，将目标分解为多个目标或准则，进而分解为多指标（或准则、约束）的若干层次，通过定性指标模糊量化方法算出层次单排序（权数）和总排序，以作为目标（多指标）、多方案优化决策的系统方法。经过我们的讨论，最终选择**层次分析法（AHP）**作为我们此次的分析模型。

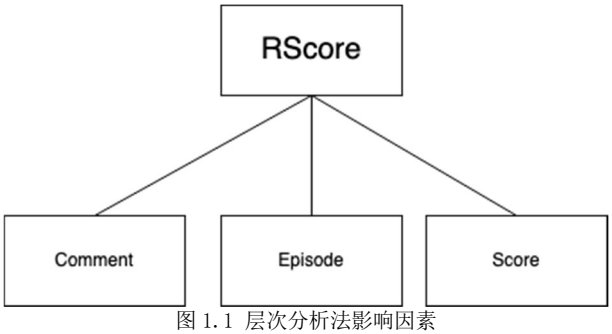


图 1.1 层次分析法影响因素

电视剧评论的数量实际上是观众数量的反映，因此我们认为**评论数量**代表观众评级，**评价分数**代表电视剧的质量，**剧集数量**代表特定电视剧制片厂的制作成本。
首先，我们排除了所有未评分的戏剧，因为如果没有最关键的评价因素，我们就无法评判电视剧。 所谓的“精彩且具有成本效益的电视剧”无疑是基于评论数量、高分和低预算的。

·评价分数

电视剧的评价分数是评论者所给的分数的均值，是给定 R 分数的主要基础。 我们将其权重值定义为 1，这恰到好处，因为分数作为首要因素，直接反映了观众对电视剧的印象。

·评论数量

电视剧评论的数量决定了它的受欢迎程度。 在某种程度上，由于虚拟网络和现实世界的讨论指数飙升，而不是其质量，电视剧在一段时间内变得如此火爆。 这种现象往往导致这样一个事实，即如果我们只是将评论和分数的数量纳入 AHP，我们将获得不准确的后果。 因此，我们将评论最多的电视剧的评论权重值定义为 1，通过这个类比，我们分别将评论数量转换为百分比，并将其添加到 R 评分的计算中。

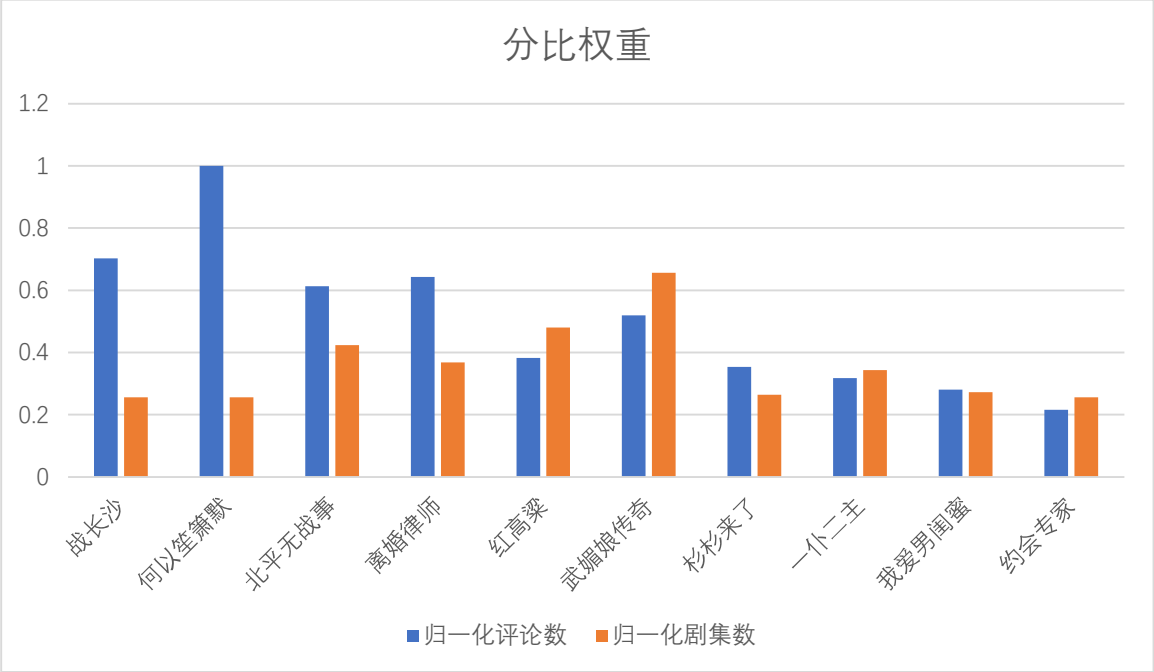
·剧集数量

众所周知，电视剧集越多，制作量就越多投资这部电视剧会花很多钱。 根据我们的第一个假设，不同系列中的每个剧集都有相同的成本，也就是说，电视剧集越多，电视剧的成本效益就越低。 我们通过设置适度的权重值 0.2 来重新定义这个因素，该值对结果的影响相对较小。
通过评价分数的计算公式我们可以知道，对于所有评论所给定的单一的评价分数取平均值即为最终的评价分数。因此我们对**评价指数** RScore 指定一个合理的计算公式：

$$\mathrm{RScore} = \mathrm{S} * \mathrm{Chs} - 0.2 * \mathrm{Ehs}$$

5.1.2 基于数据处理的实验结果分析

我们通过计算得到前十名的电视剧，并将其分比权重绘制成为柱状图，以下为绘制结果：



以下为我们所给出的前十名的人气电视剧排行榜，经过与目前各大主流视频网站排行榜进行对比，我们可以发现，这个排行榜与其有着很大的相似性，可以在一定程度上说明此模型在对于电视剧排行榜预测角度具有一定的可行性。

电视剧	评价分数	评论数量	剧集数	归一化评论数	归一化剧集数	评价指数
战长沙	9.2	18419.0	32	0.702399	0.256	6.410868
何以笙箫默	6.3	26223.0	32	1.000000	0.256	6.248800
北平无战事	8.8	16084.0	53	0.613355	0.424	5.312721
离婚律师	7.2	16872.0	46	0.643405	0.368	4.558913
红高粱	7.5	10028.0	60	0.382412	0.480	2.772093
武媚娘传奇	5.4	13625.0	82	0.519582	0.656	2.674543
杉杉来了	6.9	9285.0	33	0.354078	0.264	2.390342
一仆二主	7.2	8337.0	43	0.317927	0.344	2.220274
我爱男闺蜜	7.4	7373.0	34	0.281165	0.272	2.026224
约会专家	8.3	5664.0	32	0.215994	0.256	1.741547

表 2 前十名电视剧排名

5.2 问题二模型建立与求解

5.2.1 指标选择

为了确保数据实时性，我们的数据来源是一年内互联网上的明星公开的搜索数据

为了确保数据来源的多样性，我们的数据来源由百度搜索量等社会影响热度和微博粉丝数、百度指数等明星日常粉丝社区活跃热度两部分组成，综合分析明星的真实影响力和热度

5.2.2 模型提出（PCA 主成分分析法）

5.2.2.1 模型分析

对于图 3.1.1 中所给出的数据模型，我们有以下分析：

虽然每个指标所针对的项目不是一致的，但不同指标之间的相关性往往会对一个明星的人气指数起到一个主导性的作用。为了找到相关性和分布权重，本文引入主成分析法进行分析：

数据标准化处理：

我们使用 z-score 标准化(zero-mean normalization)对数据进行标准化操作：

$$\widetilde{S_{ij}} = \frac{S_{ij} - \overline{S_i}}{S_i}$$

我们通过这种方式可以消除不同指标维度之间的影响，并且在归一化过程中，不会影响到数据之间的相关性。

计算归一化数据的相关系数矩阵，找出特征值和特征向量：

$$r_{ii'} = \frac{\sum_{k=1}^{14} \widetilde{S_{ik}}\widetilde{S_{i'j}}}{14-1}(i, i' = 1,2,3\ldots,14)$$

最终的特征矩阵为 $R = (r_{ii'})_{14\times14}$，$r_{ii} = 1$，$r_{i'i} = r_{ii'}$，我们所给定的特征值$\lambda_i\ (i = 1,2,\ldots,14)$以及所给定的向量$L_i\ (i = 1,2,\ldots,14)$。

计算贡献率$T_k = \frac{\lambda_i}{\sum_{i'=1}^{14}\lambda_{i'}}$ 以及累计贡献率$D_k = \sum_{i'=1}^{k} T_{i'}$ 选定计算结果 $D_k \geq 85\%$的$\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_x (x < 14)$

获取每个指数在受欢迎度指数上的权重：

计算贡献率D_k并且得出T_i所对应的T_i'作为新权重，这将是我们所需要得到的主成分的权重。

5.2.2.2 模型求解

通过 5.2.2.1 章节所得出的，影响人气的几个重要指标有贴吧热度、微博热度、网络搜索量、粉丝数量和影视热度。以上的五个主要参数可以作为长期衡量指标。

最终计算所得出的明星人气指数的计算公式以及权值给出如下：

$$P_1 = 0.2806m_1 + 0.2231m_2 + 0.1804m_3 + 0.0407m_4 + 0.0942m_5 + 0.2010e^{-0.041d}$$

5.2.3 基于数据处理的实验结果分析

最终根据此模型所给出模型，我们通过使用 Python 对于目前各大主流的贴吧平台、视频平台以及搜索引擎等进行数据收集，最终套用模型得出以下的人气排行榜：

排行	姓名	排行	姓名
1	杨紫	8	赵本山
2	刘涛	9	梅婷
3	赵丽颖	10	黄河
4	陈伟霆	11	金晨
5	韩雪	12	杨颖
6	靳东	13	霍建华
7	林心如	14	…

表 3 检验明星人气排名

我们将所得到的排位与当天的官方所发布的数据进行对比，其均方差均不超过 50，对于前 20 名的均方差小余 20，所以该算法所提出的明星人气模型是能够得出可接受结果的。

5.3 问题三模型建立与求解

5.3.1 模型提出（逐步回归法）

5.3.1.1 模型分析

通过分析，我们可以看到影响电视剧受欢迎程度的指标很多

明星	影视剧类型
制作团队	拍摄成本
播出时间	播放频道
地域因素	受众人群
特效质量	…

表 5 影响电视剧受欢迎程度的因素

由于影响电视剧受欢迎程度的指标很多，可以采用逐步回归的方法选择最显著的指标。通过筛选自变量，自变量个数越大，回归平方和越大，残差平方和越小，回归分析质量越高，可以有效提高回归模型分析的精度。

5.3.2 模型求解

1）对索引重新编号并标准化不同维度：
我们首先将$y_\alpha = x_{\alpha k}$，数字的下标为$k-1$，所以数学模型为：

$$x_{\alpha k} = \beta_0 + \beta_1 x_{\alpha 1} + \beta_2 x_{\alpha 2} + \beta_3 x_{\alpha 3} + \cdots + \beta_{K-1} x_{\alpha K-1}$$

$$\alpha = 1,2,\ldots,n(\text{n 为指标数量})$$

$$S = \sum(x_{\alpha k} - \overline{x_k})^2，\ S_Q = S - S_U = \sum(x_{\alpha k} - \widehat{x_k})^2$$

此外，x_j的偏平方回归和是：

$$S_U{}' = \frac{b_j}{C_{jj}}$$

b_j为x_j的偏回归系数，C_{jj}是L^{-1}的矩阵对角线值。
2）以新指数为参数的回归模型为：

$$\widehat{x_k} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_{k-1} x_{k-1}$$

将初始数据转化为标准化回归数学模型，求解得到:

$$z_{\alpha j} = \frac{x_{\alpha j} - \overline{x_j}}{s_j}$$

并且：

$$\overline{x_j} = \frac{1}{n}\sum_{\alpha=1}^{n} x_{\alpha j}$$

3)我们可以以此得到：

$$S_j = \sqrt{l_{jj}} = \sqrt{\sum(x_{\alpha j} - \overline{x_j})^2}$$

4)我们最终可以得到初始回归曲线模型和相关性系数。建立相关系数矩阵：

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1\ k-1} \\ r_{21} & r_{22} & \cdots & r_{2\ k-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k-1\ 1} & r_{k-1\ 2} & \cdots & r_{k-1\ k-1} \end{pmatrix}$$

通过上述公式求解，我们可以得到电视剧的一些重要指标以及相应的权重。将这些指标作为参数，标准化后放入公式中，避免了单元的大小，使系统误差最小化。指数的权重与公式中的系数正相关，这意味着假设总权重为 1，权重呈多重关系；所有权重之和为 1，在最终的流行性判断公式中，可以轻松计算出相应的权重。求解得到电视剧受欢迎程度的最终公式是：

$$\boldsymbol{P_2 = 0.3634x_1 + 0.2184x_2 + 0.1723x_3 + 0.1018x_4 + 0.0102x_5 + 0.0778x_6}$$

通过这些数据，我们可以计算出不同类型的流行程度，并以数字的形式在公式中表示出来。广播频道和时间、特效水平和宣传力度可以用同样的方法计算。另外，明星人气指数见第 5.2 节。在 R 中，总权重不是 1，因为某些指标对整个指标的贡献很小。这里我们可以忽略这些索引。

5.3.3 基于数据处理的实验结果分析

为了获得理想的生产团队的名单，我们必须考虑各种指标之间的搭配。因为理想的团队不是由最受欢迎的明星、最受欢迎的类型和其他最受欢迎的指数组成的简单团队。例如，一个明星不适合所有类型的戏剧，而一个团队通常在特定类型中表现出色。应该建立一个新的矩阵来描述搭配关系。

$$C_i = [C_{i1}, C_{i2}, C_{i3}, C_{i4}, \dots, C_{in}]$$

C_i为第 i 个明星的特征矩阵，其中包含了与该明星所参与的各种戏剧的搭配程度。n 是该明星出演戏剧的数量。当与戏剧类型匹配时，匹配系数$\delta = C_i \times D_i$，D_i是戏剧类型的特征矩阵。匹配系数越大，匹配效果越好。同时，制作团队要像明星一样打字。每个索引都可以用一个特定的矩阵表示。结合公式，我们可以得到如下理想团队：

导演	孔笙
明星	赵丽颖、胡歌、白敬亭、杨紫
类型	当代城市剧
策划	王丽萍、海岩
组织	山东电影电视剧制作中心

表 6 模型预测最佳团队

最受欢迎的戏剧类型是当代都市剧，这可以作为一个基本指标。结合公式，我们可以找到最匹配的明星、导演和编剧。

5.4 问题四模型建立与求解

5.4.1 模型介绍（PUM-CF 算法）

基于改进谱聚类的协同过滤算法(PUM-CF)，相比于一些经典的聚类算法，谱聚类有自己独特的优点且已被很多领域所使用，它首先求解出图的拉普拉斯矩阵的特征值和特征向量，把原来的数据聚类问题转换为特征向量的聚类问题，通过对拉普拉斯矩阵特征向量的聚类完成原始数据的操作。所以谱聚类既可以对简单信息进行处理，还可以有效地对高维复杂数据进行聚类操作。

5.4.1.1 模型分析

在推荐系统中，一个用户能够拥有很多个兴趣，同时也可以归于多个分组，谱聚类算法要能够实现数据模糊聚类的实际要求。

谱聚类首先求解出图的拉普拉斯矩阵的特征值和特征向量，把原来的数据聚类问题转换为特征向量的聚类问题。一般使用 K-means 进行后续的聚类操作，初始簇心设置的是否合适会严重影响最终的结果，所以要改善谱聚类初始值敏感的问题。

数据点之间的相似度计算准则也对聚类的准确度造成了很大的影响，因此要依据具体的应用场景来使用合适的计算准则。

针对上述问题，以下引出基于最大距离积的改进谱聚类算法，把该改进的谱聚类算法引入到推荐系统中。

5.4.2 模型提出

用户对具有一些明显特征属性标签物品进行评价间接的说明对这一类别的物品的偏好，例如用户经常性评价动作类的电视剧，那么也就从侧面说明了用户有可能对动作类的电影有偏好。

根据这种现象的存在，将物品的特征标签与用户对一些特定属性标签的评价记录进行融合，从而得到了用户对于不同特征属性物品的评价矩阵。我们定义了用户对于某些特征属性物品的用户偏好矩阵 UP:

$$P(u_k) = \frac{\sum_{ie I_{u,x}} R_{u,i}}{sum(I_{u,x})}$$

上式中用户 u 评论过的物品里拥有 x 特征标签的物品集合记为$I_{u,x}$，用户 u 对于物品 i 的实际评分是使用$R_{u,i}$来表示，$I_{u,x}$中的项目总数记为$sum(I_{u,x})$。如下表所示是拥有 U_1, U_2, U_3及项目 I_1, I_2 ,I_3的 user-items 评分矩阵：

	I_1	I_2	I_3
U_1	3	4	4
U_2	*	2	*
U_3	4	*	3

表 7 user-items 矩阵

用户偏好矩阵 UP 能够使用下面的几个步骤来计算：

- 1、得到用户对于项目的评分信息。
- 2、得到项目的特征属性类别数据。
- 3、获得各个用户对于不同特征物品的偏爱度。

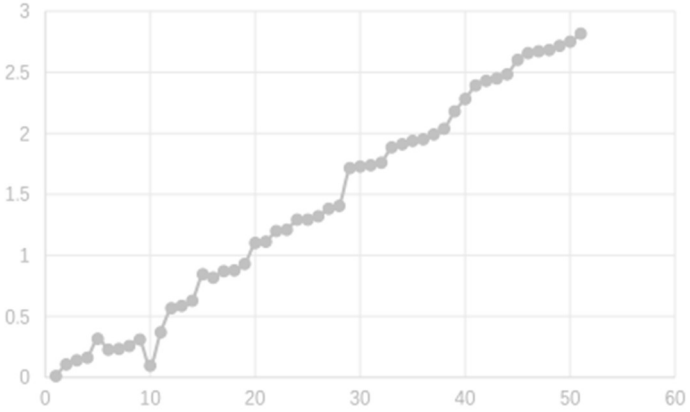
因此能够得到评分矩阵 S 和项目特征属性 P 得出用户偏好矩阵 UP 如下表所示：

	T_1	T_2	T_3	T4
U_1	3.5	3.67	3.5	5
U_2	0	2	2	2
U_3	3.5	3.5	4	0

5.4.3 基于数据处理的实验结果分析

为了验证模型的可信性，构建了一个新的模型来测量。通过计算平均值和误差的标准偏差。

关于误差，我们根据指数计算结果搜索有多少类型与感兴趣的类型的不匹配值以及获取其的数量，将模型后的残差数值相加，运行数次并与输出的全部数据进行比较。然后计算平均值和标准差。



最终，可信度在 89.28%到 97.22%之间，平均可信度为 93.2%，这意味着该模型可以良好的胜任对于智能推荐算法的应用。

6. 模型的评价

6.1 模型优点

• 我们的模型数据来源于互联网，其具有客观和高通用性，通过真实情况所获取的数据设计的模型与官方网站所给出的排位信息相符合，很好的证明了我们的模型的通用性。通过模型验证我们可以看到我们的模型错误率低于可接受范围，证明了我们的模型有着很好的正确性。

• 数据来源丰富并且来源于多种渠道，多个平台，使得我们的模型具有很好的稳定性。

• 通过我们的 PUM-CF 算法得出的用户偏好矩阵不仅能够应用在用户的影视喜好智能推荐上，还可以应用在其他相关领域之中。

• PUM-CF 算法只依赖用户行为，无需对内容进行深入了解，适用范围广

• PCA 主成分析法使得数据集更易使用，降低算法的计算开销

6.2 模型缺点

- （1）数据来源广泛但是缺乏验证，这些数据来源缺乏多次有效地验证，也许这一部分在模型中的权重影响比较大。•所有数据并非来源于官方，依赖于平台发布的数据，可能会影响模型的精度。
- （2）问题一中的 AHP 层次分析法是一种带有模拟人脑的决策方式的方法，因此必然带有较多的定性色彩。
- (3) 问题四中的 PUM-CF 算法一开始需要大量的<user,item>行为数据，即需要大量冷启动数据且很难给出合理的推荐解释

7.参考文献

[1]Zhou Ya,The TOPSIS in the Multiple Attribute Decision Making, Wuhan Universityof Technology,2009
[2]Liu Sifeng, Cai Hua, Yang Yingjie,Cao Yin, The research progress of GRA, Instituteof Gray System, Nanjing University of Aeronautics and Astronautics, 2013,8
[3]Hongping Zhao ,Usage of Stepwise Regression based on Different econometricssoftware,School of Economics,Law and Politics,Nanjing Xiaozhuang University,2007,09
[4] 秦晓阳. 基于聚类的智能推荐算法研究及应用[D].电子科技大学,2018.
[5] PCA 主成份分析方法 https://www.cnblogs.com/haore147/p/3630002.html

附录

数据预处理

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt

part = 2
if part == 1:
    dt1 = pd.read_excel("附件一-电视剧评估信息.xlsx",'Sheet1')
    dt2 = pd.read_excel("附件二-电视剧基本信息.xlsx",'Sheet1')

    writer = pd.ExcelWriter("result.xlsx")
    #print(dt1.merge(dt2,left_on='TV Drama',right_on='TV Drama'))
    dtret = dt1.merge(dt2,left_on='TV Drama',right_on='TV Drama')
    dtret.to_excel(writer,'Sheet1')
    writer.save()
    writer.close()
    classes = {}
    for i in dt1.groupby('Theme'):
        classes.update({i[0]:i[1]})
    print(str([i for i in classes.keys()]))
    cnt_drama = [0 for i in range(12)]
    cnt_drama_comments = [0 for i in range(12)]
    for   i,j   in   zip(dtret.sort_values(by='Date   of   Issuance
License')['Date of Issuance License'].reindex(),
                  dtret.sort_values(by='Date      of      Issuance
License')['Number of Comments'].reindex()):
        cnt_drama[int(i.split("/")[1])-1]+=1
        cnt_drama_comments[int(i.split("/")[1])-1]+=j
    #print(dtret[dtret['Score'] == 0])
    month = [i for i in range(1,13)]
    plt.bar(month,cnt_drama)
    plt.show()

dt = pd.read_excel('result.xlsx','Sheet1')
comment_max = dt['Number of Comments'].max()
Score_max = dt['Score'].max()
Episode_max = dt['Episode'].max()
Info_total = []
for i in dt.index:
```

```
    Info_new = []
    for j in dt.columns:
        if j == 'TV Drama':
            Info_new.append(dt[j][i])
        if j == 'Score':
            try:
                a = float(dt[j][i])
            except:
                a = 0
            Info_new.append(a)
        elif j == 'Number of Comments':
            try:
                a = float(dt[j][i]/comment_max)
            except:
                a = 0
            Info_new.append(dt[j][i])
            Info_new.append(a)
        elif j == 'Episode':
            try:
                a = float(dt[j][i]/Episode_max)
            except:
                a = 0
            Info_new.append(dt[j][i])
            Info_new.append(a)
    Info_total.append(Info_new)
for i in Info_total:
    Rscore = i[1]*i[2] - 0.2 * i[3]
    i.append(Rscore)

Info_dt = pd.DataFrame(Info_total)
Info_dt.columns = ['TV Drama','Score','Number of
Comments','Chs','Episode','Ehs','RScore']
print(Info_dt.sort_values(by='RScore',ascending=False).head(10))
```

明星筛选

```
import os
import shutil
import re
from bs4 import BeautifulSoup
```

```
listdir = os.listdir(os.getcwd())
if not os.path.exists("hm"):
    if not os.path.isdir("hm"):
        os.makedirs(os.path.join(os.getcwd(),"hm"),mode=0o777)
        print("Success create the html folder")
    else:
        print("Warnning: Html dir is already existed!")
else:
    print("Warnning: Html is already existed!")

try:
    for name in listdir:
        if re.match(r'.*_hm.txt',name) is not None:
            shutil.move(name,"hm/")
            print (name + "-->" + os.getcwd()+"/hm/"+name,end="\n")
    print("Move files successfully")
except:
    print("Already done!")
    pass

stars_bd_file_list = os.listdir("html")
for file in stars_bd_file_list:
    try:
        with open(os.path.join(os.getcwd(),"html",file),"r") as f:
            line = 0
            try:
                for i in f.readlines():
                    line+=1
            except:
                pass
            print("length:"+str(line))
            if line <= 10:
                f.close()
                os.remove(os.path.join(os.getcwd(),"html",file))
                continue
            bsdt                                                    =
BeautifulSoup(f.read(),"html.parser",from_encoding='utf-8')
            try:
                for                          i                          in
bsdt.find(attrs={"class":re.compile("hint_[a-zA-Z0-9]{0,9}
c_font_[a-zA-Z0-9]{0,9}")}):
                    print(i)
```

```
            except:
                print(file)
                pass
    except:
        print("utf-8"+file)
```

百度信息搜集

```
import requests
import time
import pandas as pd
import numpy as np
import os
import json

name = "白雪"
url                                                                    =
r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'
+name+'","wordType":1}]]&area=0&days=30'
print(url)
headers = {
"Host": 'index.baidu.com',
'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0)
Gecko/20100101 Firefox/94.0',
'Accept': 'application/json, text/plain, */*',
'Accept-Language':              'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-
US;q=0.3,en;q=0.2',
'Accept-Encoding': 'gzip, deflate, br',
'Referer': 'https://index.baidu.com/v2/main/index.html',
'Connection': 'keep-alive',
'Cookie':                   'BAIDUID=6EE4B14CF0C5FFBD8BC51FE2BD5CC1A2:FG=1;
BIDUPSID=6EE4B14CF0C5FFBDCDDAA3478440FB86;                  PSTM=1628165068;
__yjs_duid=1_6bfdb0ac0e3760480bbcac4da6e828871628176157265;
BDUSS=X5NT1F4RjA2UmN4WnBOaVNkdX5QOC1iM2Nad3kyMmEzVUFmMmgxM1AzRUw4Mk5oRVFB
QUFBJCQAAAAAAAAAAAEAAACPk8sQt67U88jwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAtmPGELZjxhV;
BDORZ=FFFB88E999055A3F8A630C64834BD6D0;
H_PS_PSSID=35295_35104_31254_35435_35456_34584_34504_34606_35329_35315_26
350;                                        BDSFRCVID=hFtOJexroG04sxcH4wwF-
tse4ec_23JTDYLEL5mveY16oVFVJeC6EG0Ptq6Ex2P-
EHtdogKKy2OTH9DF_2ux0jjg8UtVJeC6EG0Ptf8g0M5;
H_BDCLCKID_SF=tbAOVCD2JKL3fP36q4OHht_tMq0X5-RLf273ah7F5l8-hCb6b-
```

```
IKXUv3etoMeR3e5RnB3qrCBD3xOKQphUIhyjO-
O4FLBfOqbCJLLnbN3KJm8tP9bT3vLfuyjtcz2-
biW5KH2MbdflbP_IoG2Mn8M4bb3qOpBtQmJeTxoUJ25DnJhbLGe6L3-RJH-
xQOKnLXKKOLVKDyW17ketn4hUt2jbtb-PAJ-
bbjH6neOCbHKMAKhpc2QhrdQf4WWb3ebTJr32Qr-qcHOKQpsIJM5bL-QJvXO2c4Bn3baKvia-
I-BMb1bqbDBT5h2M4qMxtOLR3pWDTm_q5TtUJMeCnTDMFhe6oM-Uvy-
4OXKD6OOPK8Kb7VbnnD5MnkbJkXhPtjtnJOK2-JL-3IH4b5eDbN-pPKDUI7Qbrr0xAta6c-
hR4KaMQPS1cNLTjpQT8r5hLqORkHLPj4aPbeab3vOPI4XpO1ef0zBN5thURB2DkO-
4bCWJ5TM15jDh3Mb6ksD-FtqtJHKbDH_KPMtMK;
ab_sr=1.0.1_Mjk1MjAzY2E5NTZkODAxY2NiNDQ5YmY4N2U5NzExOWNmZmQOOWE3YTM4ZGUxM
mI1MzNjMTYxMDIxMWQxZWU3NmU5MTF1Yzc2NTQOMmQzYjZmZDVjNzQwMzE5NjQ4Yjk2ZjFmMm
I3NGJhMGQ4NT1mN2FiZmMyOWIwYzgxZTIONjAwMjkyYTB1ODA3ODU2NjNkN2QyNTRhMDc1M2E
OM2FhMA==;  BA_HECTOR=a021248020058524cd1gr6nok0r;  delPer=0;  PSINO=6;
BDRCVFR[gltLrB7qNCt]=mk3SLVN4HKm;
Hm_lvt_d101ea4d2a5c67dab98251f0b5de24dc=1639145239;
Hm_lpvt_d101ea4d2a5c67dab98251f0b5de24dc=1639145471;
RT="z=1&dm=baidu.com&si=6rc7piqj5be&ss=kx0gr626&sl=d&tt=6o8&bcn=https%3A%
2F%2Ffclog.baidu.com%2Flog%2Fweirwood%3Ftype%3Dperf&ld=5109";
__yjs_st=2_ZmE0MWQ4MjVkYmQwMzcyYWZlNGYzNDY0MGFlMGNlMTdiZTBmZDIyNzNhZTQyNW
MONWQ1OTR1ZDE0MjFlNzhhMWU5OWUyZDdmZjE3MTgxNjBhOWY5YTNiMGQxYjIzYTU5ZDRmOWR
hMTgyOWFkODFkOGMzOTQ0NTE5N2IxZGFmODY1YzUzM2NiZDUwOGE2YTVkMzQ5Yz1jMDk2Y2Rk
ZmQwY2QwNDVhNTNmNmYyZWQ1Zjk4YWMzYTRiODM4NDAzMjNhNDd1ZmUwMjE4N2M4YjY3Y2JkM
zMzMjB1OWZhMDQ1YzkOYmNmYz1kM2EwZWU3NjA0Zjg5YTgzNGMwZGQ2NWE3MmJkMzYxNGRiNj
Z1ZDJiOGM2NGMwNzAwNzgxZWZ1MzMzXzdfNDNkM2FjNmQ=;
bdindexid=vo1c73ivop6an83hkaavraog13',
'Sec-Fetch-Dest': 'empty',
'Sec-Fetch-Mode': 'no-cors',
'Sec-Fetch-Site': 'same-origin',
'Pragma': 'no-cache',
'Cache-Control': 'no-cache'
}

dt = pd.read_excel('result.xlsx','Sheet1')
stars = set()
for names in dt['Starring']:
    if names is not np.nan:
        for  name  in  str(names).replace('  、  ',',').replace('  ,
',',').split(','):
            stars.add(name)
total_file = open("summary_hm.txt","a+")
total_file.seek(0,0)
for name in stars:
```

```
    time.sleep(1)
    try:
        if os.path.exists(name+'_hm.txt'):
            continue
        with open(name+'_hm.txt','w') as f:
            time.sleep(1)
            url                                                    =
r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'
+name+'","wordType":1}]]&area=0&startDate=2021-11-10&endDate=2021-12-09'
            print(url)
            r = requests.get(url = url,headers = headers)
            r.raise_for_status()
            r.encoding = r.apparent_encoding
            ret = str(r.text)
            ret = json.loads(ret)

print(name+','+str(dict(ret['data'])['index'][0]['generalRatio'])+'Succes
s')

f.write(name+','+str(dict(ret['data'])['index'][0]['generalRatio'])+'\n')

total_file.write(name+','+str(dict(ret['data'])['index'][0]['generalRatio
'])+'\n')
    except:
        print(name + "Failure")
        continue

# for name in stars:
#     time.sleep(1)
#     try:
#         if os.path.exists(name+'_hm.html'):
#             print(name,"Success")
#             continue
#         with open(name+'_hm.html','w') as f:
#             time.sleep(1)
#                                                                 url     =
r'http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=[[{"name":"'
+name+'","wordType":1}]]&area=0&days=30'
#             r = requests.get(url = url,headers = headers)
#             r.raise_for_status()
#             r.encoding = r.apparent_encoding
#             ret = str(r.text)
```

```
#             print(ret)
#             f.write(ret)
#             print(name,"Success")
#     except:
#         print(name,"Failure")
#         continue
```

新浪微博信息收集

```
import pandas as pd
import numpy as np
import os
import time
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
import requests
from selenium.webdriver import Chrome
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver import *
from selenium import webdriver
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.support.wait import WebDriverWait
from urllib.parse import quote
from pyquery import PyQuery as pq

dt = pd.read_excel('result.xlsx','Sheet1')
stars = set()
for names in dt['Starring']:
    if names is not np.nan:
        for name in str(names).replace('  、  ',',').replace('  ,',',').split(','):
            stars.add(name)

url = "https://s.weibo.com/weibo?q="
headers = {
"Host": "s.weibo.com",
"Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8",
"Accept-Language":              "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-
```

```
US;q=0.3,en;q=0.2",
"Accept-Encoding": "gzip, deflate, br",
"Connection": "keep-alive",
"Cookie":                "www.52jingsai.com,widget.weibo.com,www.baidu.com;
SUBP=0033WrSXqPxfM725Ws9jqgMF55529P9D9WWbjVd7EgcvSe75XM2kBhOX5JpX5KMhUgL.
Foe4eh2pSo.7SO-2dJLoIp7LxKML1KBLBKnLxKqL1hnLBoM01K5peKq4ehMf;
SINAGLOBAL=8505429497315.529.1632922843183;
ULV=1639136383200:2:1:1:3763013819565.094.1639136383093:1632922843184;
ALF=1670672351;
SCF=AoMImLiEPADgRCx1ffu6aj6AK92GFULp5oaCrYYu3UyFQ1CaK7GE81ZnOmRTsnXxymUak
y46S2SrG5WmaKwfqDc.;
SUB=_2A25MtOwwDeRhGeVH61MQ9ifMzDmIHXVvxTr4rDV8PUNbmtAKLUGtkW9NT2jo5JhLkwN
EFUwWQ_iTWmiOS-eqHHhv; SSOLoginState=1639136352; _s_tentry=www.baidu.com;
Apache=3763013819565.094.1639136383093",
"Upgrade-Insecure-Requests": "1",
"Sec-Fetch-Dest": "document",
"Sec-Fetch-Mode": "navigate",
"Sec-Fetch-Site": "cross-site",
"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0)
Gecko/20100101 Firefox/94.0",
}
# for name in stars:
#     time.sleep(1)
#     try:
#         if os.path.exists(name+'_sina.html'):
#             continue
#         # with open(name+'_sina.html','w') as f:
#         time.sleep(1)
#         r = requests.get(url = url+name,headers = headers)
#         r.raise_for_status()
#         r.encoding = r.apparent_encoding
#         ret = str(r.text)
#         print(ret)
#         #f.write(ret)
#     except:
#         continue

for name in stars:
    time.sleep(1)
    try:
        if os.path.exists(name+'_sina.html'):
            print(name,"Success")
```

```
            continue
        with open(name+'_sina.html','w') as f:
            time.sleep(1)
            r = requests.get(url = url+name,headers = headers)
            r.raise_for_status()
            r.encoding = r.apparent_encoding
            ret = str(r.text)
            # print(ret)
            f.write(ret)
            print(name,"Success")
    except:
        print(name,"Failure")
        continue
```

新浪微博文件处理

```
from bs4 import BeautifulSoup
import os
import re

list_file = os.listdir(os.getcwd())
for file_name in list_file:
    if re.match(r'.*._sina.html',file_name) is None:
        list_file.remove(file_name)
total_fans_cnt = open("total_fans_cnt.txt","a+")
total_fans_cnt.seek(0,0)
for file in list_file:
    try:
        with open(file,"r") as f:
            name = file.split('_')[0]
            bsdt                                                    =
BeautifulSoup(f.read(),"html.parser",from_encoding="utf-8")
            if  bsdt.find("span",attrs={'class':'s-nobr'})  is  not
None:
                print(name+"
"+str(bsdt.find("span",attrs={'class':'s-nobr'}).contents[0]))
                total_fans_cnt.write(name+"
"+str(bsdt.find("span",attrs={'class':'s-nobr'}).contents[0])+'\n')
    except:
        continue
print(list_file)
```

明星排位

```
import os
import re
import json
import ast
filelist = os.listdir(os.getcwd())
for i in filelist:
    if re.match(r'.*_hm.txt',i) is None:
        filelist.remove(i)

total = []
for file in filelist:
    with open(file,"r") as f:
        for line in f.readlines():
            splited = line.split(",")
            str1    =    "".join(','+splited[i]    for    i    in
range(1,len(splited)))
            ret     =     ast.literal_eval(str(str1[1:]).replace("
","").replace('\n',''))
            ret.update({'name':splited[0]})
            print(ret)
            total.append(ret)

for i in sorted(total,key=lambda x:x['avg'],reverse=True)[:34]:
    print(i['name'])
```