《商务数据分析与应用》课程期末报告

银幕下的数据回响

——电影评论的多维分析与应用

何柔叶　学号：20211357　专业班级：信息 2102

目录

1、选题背景及问题提出 3
商务数据分析选题的实际背景 3
问题的提出及实际意义 4
2、商务数据分析的技术路线 5
商务数据分析的过程、步骤 5
1.数据采集 5
1. 八爪鱼采集器的使用： 5
2. 数据采集的限制和解决方法： 6
3. 从网上下载数据集： 7
4. 数据集的选择（最终选择只使用 Kaggle 上面下载的数据集）： 8
2.数据预处理 9
1.原始数据集的分析 9
2.分析数据预处理的步骤 9
3.删除重复值 9
4.删除缺失值 10
5.筛选重要字段 10
6.统一字段的数据格式 10
7.过滤垃圾评论 10
3.数据分析 11
4.数据挖掘 13
采用的方法和技术 14
1. 八爪鱼采集器: 14
2. 利用 Kaggle 网站进行数据集下载: 15
3. Python: 15
4. Jupyter Notebook: 15

5. Scikit-learn (随机森林模型): 15
6. LDA 主题建模: 15
7. 自定义文本挖掘算法: 15
8. Pyecharts: 15
9. Matplotlib 和 Seaborn: 16
10. Pandas: 16
11. NumPy: 16
12. 镝数图表: 16
3、商务数据分析工具 16
数据采集采用的方法和技术 16
1. 八爪鱼采集器: 16
2. Kaggle 数据集下载: 17
数据预处理采用的方法和技术 17
数据统计分析采用的方法和技术 17
数据挖掘采用的方法和技术 18
4、商务数据分析过程 18
4.1 数据描述 19
八爪鱼采集器采集到的数据： 19
网上下载的数据： 21
预处理后的数据： 22
4.2 数据分析过程 23
数据统计分析（以绘制各个电影的垃圾评论平均评分、所有评论的平均评分三维柱状图为例） 24
1.导入所需库和数据： 24
2.计算垃圾评论的平均评分： 24
3.计算所有评论的平均评分： 24
4.合并两个数据集： 24
5.查看合并结果： 25
6.然后，我们使用这个数据集，使用 pyecharts 来绘制图形。具体操作如下： 25
数据挖掘分析 26
4.3 结果分析 27
1.数据采集的过程及结果 27
2.数据预处理的过程及结果 31
3.数据分析的过程及结果 34
4.数据挖掘的过程及结果 35

5.数据统计分析采用的方法、必要的软件运行界面截图、关键程序代码、结果截图或列表等。 37
6.数据挖掘采用的模型或算法、建模过程、算法参数设置等以及必要的软件运行界面截图、关键程序代码、结果截图或列表等。 40
一、基于随机森林模型的评论情感倾向预测 40
二、基于 LDA 主题模型的评论分类预测 43
三、基于自定义文本挖掘算法的受欢迎的角色、元素倾向统计 48
5、结果分析与讨论 53
基于观众喜好的个性化推荐 53
垃圾评论的管理与过滤 53
电影宣传策略优化 53
观众理性看待评分的建议 54
电影发布时间建议 54
基于观众评价的电影改进意见 54
海报、周边设计、后续作品演员选择 54
垃圾评论的影响 54
评论数量的变化 55
评论的统计性分析 55
电影热度分析 55
电影行业趋势分析 55
电影持续影响力分析 56

1、选题背景及问题提出

说明商务数据分析选题的实际背景，所提出的商务数据分析问题，以及所提出问题的实际意义。

商务数据分析选题的实际背景

在当今社会，电影行业已经成为了一个重要的文化和经济领域。随着科技的发展和社交媒体的普及，观众通过网络平台分享电影体验已成为常态。这些在线评论不仅反映了观众的直接反应，也成为电影制片方、发行商和市场分析师了解电影市场反

馈的重要途径。观众评论中蕴含的丰富信息，如情感态度、喜好趋势和观影体验等，对于指导电影制作、营销策略和提升服务质量具有重要意义。

虽然传统的电影评价方法（如票房、专业评论和观众评分）能在一定程度上反映电影的受欢迎程度，但它们通常无法深入挖掘观众的具体情感和详细喜好。随着文本挖掘技术的发展，我们现在有机会更深入地探索评论中的信息，以洞悉观众的真实反馈。

问题的提出及实际意义

在这个背景下，我们的问题提出源于对电影评论潜在价值的发现。注意到电影评论中包含丰富的信息，如情感倾向、评分影响、垃圾评论识别等。如果能有效挖掘这些信息，就能为电影营销策略提供支持，并提升评论系统的质量。

我们选择了对电影评论数据进行深入分析的商务数据分析项目。我们的目标是通过对电影评论的系统分析，提取出观众的情感倾向、喜好趋势、评论中的关键元素以及识别垃圾评论等关键信息。这一分析不仅可以帮助电影制片方和发行商了解观众反馈，调整营销策略，还能为电影评价网站提供数据支持，优化用户体验和评论质量。

具体地，我们的问题提出涉及以下几个方面：

1. **情感分析：** 通过分析评论中的情感倾向，了解观众对特定电影的整体态度和情感反应。
2. **趋势分析：** 通过时间序列分析，观察评论数量和评分随时间的变化趋势，为电影的宣传时机和长期声誉管理提供数据支持。
3. **关键元素提取：** 通过提取评论中的关键元素（如角色、剧情、特效等），了解观众关注的焦点和电影的热点话题。
4. **垃圾评论识别：** 通过识别和过滤垃圾评论，提升评论数据的质量和可信度，为观众和电影评价网站提供更加健康和真实的评论环境。

对这些方面的深入分析，我们的项目旨在为电影行业的各方利益相关者提供实际和有效的市场洞察。这对于优化电影产品、提升观众满意度、增强营销效果和改善评论平台环境具有重大意义。

综上所述，本项目的核心目标是通过深入分析电影评论数据，为电影发行商、消费者、影评网站提供实时、有效的市场反馈，同时提升评论环境质量，增强用户体验。随着大数据和文本挖掘技术的应用，我们将不再只依赖于票房等表面数据，而是深入挖掘评论中的细微情感和详细意见，为电影质量的提升、观影体验的丰富和市场份额的拓展提供支持

2、商务数据分析的技术路线

总体说明商务数据分析的过程、步骤及所采用的方法和技术，可以采用流程图或框图进行说明。

商务数据分析的过程、步骤

为了回应上面的这些问题，我们需要经历数据采集、数据预处理、数据分析、数据挖掘这几个步骤。

1.数据采集

1. **八爪鱼采集器的使用**：

我负责使用八爪鱼数据采集器采集数据，这是我第一次使用这样的工具，因此刚开始有些不熟悉。

通过在 Bilibili 上查找相关教程，我学会了如何操作八爪鱼采集器。

我尝试了两种方法：模板采集法和自定义采集法。在模板采集法中，虽然采集过程快速，但数据质量不理想。因此，我转而尝试了自定义采集法，这需要更多的手动设置，但能够更精确地控制采集的数据。

在模板采集法中，我们一共找到了两个合适的模板。在使用第一个模板时，我们遇到采集效果不理想的问题，采集到的是追评，即采集的是对于评论的回复，而非初始评论。

经过分析，发现，因为豆瓣网中的每个评论都可以有追评，所以这个模板把所有的追评（及评论下方的回复）都爬取了下来，结合实际，评论的追评部分往往是网友之间互相争吵的地方，因此，这些追评并不是我们数据爬取想要的结果，而真实的、被回复的、原来的评论则不断重复，很多条数据中才会出现一条这样的评论，因此，不能使用这个模板。于是我们开始尝试第二个模板。

在使用第二个模板时，我们遇到了采集数量上限的问题，只能采集前 220 条的评论数据。然后，我们检查了八爪鱼采集器的各种设置，并查找了有关的问题，但是没有找到相关的结果，对八爪鱼的配置进行了修改也没有成功突破数据采集的数量限制。另外，这个模板采集速度过慢，因此，我们尝试使用其他方法，即自定义采集。

在自定义采集中，我学会了设置循环翻页、选取评论列表，并提取相关字段。我还发现了采集数据量的上限问题，仅能采集到前 220 条评论数据。

2. **数据采集的限制和解决方法**：

我们在采集过程中遇到了数据量限制的问题。我们似乎每个电影都只能收集前 220 条数据。为了解决这个问题，我们尝试了多种方法，排查了各种原因，可能的原因有：到底是因为豆瓣只提供了前 220 条数据可供查看？还是因为八爪鱼采集器的限制？又或者是豆瓣网上有着一定的反爬机制造成的限制?

带着这些问题，回到了豆瓣网的《长城》的评论区，如下图所示，我们选择的是看过的人的全部的评论，所以应该能够显示出所有的评论才对：

然后，我们检查了八爪鱼采集器的各种设置，并查找了有关的问题，但是没有找到相关的结果，对八爪鱼的配置进行了修改也没有成功突破数据采集的数量限制：

最终，我们发现这是因为我们没有登录账号，所以只能看到前 11 页的数据，而前 11 页的数据刚好 220 条，这恰恰说明了我们是由于没有登录账号导致采集数量受限。

随后我们登录账号，但是即使登录，也只能查看前 30 页的评论数据，一共 600 条，我们尝试解决但是无果。

我们为了扩充数据量，决定分别从好评、一般、差评中获取不重复的评论数据。这样，每部电影就可以采集到前 1800 条数据。

3. **从网上下载数据集**：

通过网上查找数据集获取的方法，我们了解到 Kaggle 作为一个知名的数据科学社区，常被数据分析师和研究人员用来分享和寻找数据集。

Kaggle 是一个著名的数据科学竞赛平台，提供了广泛的数据集资源。在获取数据集的步骤中，我们首先登录 Kaggle 网站，然后导航至 Datasets 部分。在这里，我们利用搜索功能查找到所需的电影评论数据集。

找到目标数据集后，我们进一步浏览了数据集的描述、数据量、字段信息等，以确保数据集符合我们的分析需求。确认无误后，我们下载了数据集。下载的数据集以 CSV 文件格式存储，这也便于我们后续在 Python 等数据分析工具中进行操作和分析。

4. **数据集的选择**（最终选择只使用 Kaggle 上面下载的数据集）：

由于两种方式获得的数据集有很大的差异，所以我们不能够简单的将两个数据集相整合来使用。具体原因如下：因为他们包含大量重复的数据集，而且选取的电影也基本不相同，而且从八爪鱼采集器采集到的 2017 年以后的评论数据也会因为数据量过少被稀释掉，从而难以发挥作用，同时也会导致不同电影之间，评论的数量有着很大的差异。

两种数据集的具体对比分析如下：

数据集	优点	缺点
八爪鱼采集器采集到的数据集	评论内容是最新的，电影内容也是最新的，根据豆瓣电影的分类，每一种各选了一部电影，同时也包含了中国，美国，英国等地的电影，含有我们所需的所有字段，数据预处理简单。	数据量相对过少。
从网站 Kaggle 上下载的数据集	数据量十分庞大，含有我们所需的所有字段，数据预处理简单。	评论内容和电影并不是最新的，因为爬取的时间是在 2017 年，因此数据集中只有这之前的评论和电影数据。

因此两种数据集各有优缺点

结合我们的研究目的，我们主要是需要根据评论数据来获取对于电影发行商，消费者，影评网站的建议，这需要庞大的数据量的支持，因此，从网站 Kaggle 上下载的数据集作为我们的研究使用的数据集更加合适。因此，在后续研究中，我们都使用从网站 Kaggle 上下载的数据集作为研究对象。

2.数据预处理

1.原始数据集的分析

原始数据集的信息如下所示，一共 2125056 条评论数据，一共 28 部涵盖了各种类别的电影。包含了我们所需要的全部字段，数据集的储存方式为储存在同一个 csv 文件中，一共 406MB：

我们发现，数据集中包含了很多我们不需要的字段，需要删除，

其次，日期字段的格式需要确保是能够被 python 处理的格式。星级，点赞数，评论内容，电影中文名这几个字段分别都以正确、易处理的的形式所储存（整型、字符串），因此不需要进行数据类型的转化。

另外，通过观察发现，评论数据中有一些和电影无关的内容（广告,无关评论等），我们将其称作垃圾评论,这些无关评论对数据分析的准确性造成了一定影响，因此需要对这些垃圾评论进行筛选并删除。

2.分析数据预处理的步骤

在我们的项目中，数据预处理的关键步骤包括筛选重要字段、删除重复值和缺失值、以及过滤垃圾评论。通过这些步骤，我们旨在提高数据的质量和分析的准确性，以减少数据集的大小和维度。例如，筛选重要字段是为了减少数据集的大小和维度，节省后续操作的运行时间和空间。删除重复值和缺失值是为了确保数据的完整性和代表性；而垃圾评论的过滤则是为了提高数据分析的有效性和准确性。

3.删除重复值

通过分析，我们发现数据集中存在 16 条重复评论，因此删除了其中的 8 条评论

4.删除缺失值

我们经过分析，发现数据集中不存在缺失值

5.筛选重要字段

为了减少数据集的大小和维度，加快后续操作的运行，同时节省储存空间。我们使用 python 筛选出了最重要的五个字段，即：电影名、评论日期、评论内容、星级评分、点赞数

6.统一字段的数据格式

经过分析，我们发现我们的筛选后的五个字段的数据格式都是统一的，评论日期为日期形式，电影名和评论内容为文本型，星级评分和点赞数为整数型。

7.过滤垃圾评论

- 我们的第一步是设置黑名单,并设定一个阈值,如果一个评论满足阈值数量的黑名单中的关键词，则被标记为垃圾评论
- 第二步是设置白名单,通过设置白名单，对所有第一次标记为垃圾评论的评论进行纠正，重新标记为正常评论
- 第三步是经过黑白名单得出垃圾评论结果,并查看效果
- 然后,我们经过不断改变黑白名单中的关键词,并查看效果,观察哪些黑名单中的关键词设置的不好，总是误将正常评论标记为垃圾评论，就将该关键词删除,或者继续增添白名单中的关键词，修改误标记的评论。
- 然后我们查看了阈值分别设定为 1 和 2，并不断更改关键词之后，得到的结果：可以看到阈值设定为 2 时，真正垃圾评论的占比有显著提升.但是经过该方法,筛选到的垃圾评论总量只有 406 条,而将阈值设定为 1 时,筛选到的总评论数量总有 6732 条.因此两种方法各有优劣,阈值设定为 2 筛选出的垃圾评论非

常准确,但是可能会遗漏很多垃圾评论,使他们没有被筛选出来。最终,我们采取了将阈值设定为 1 的方法，过滤掉了数据集中存在的垃圾评论。

- 然后，我们分析了黑白名单的关键词的作用性大小。
- 其中，我们将黑名单关键词的作用定义为在垃圾评论中,该关键词出现的次数。白名单关键词的作用定义为,在黑名单初步筛选得出的结果中,白名单关键词出现的次数。

经过数据预处理，数据的条数为 2118316 条，储存形式为在 python 代码的变量中，我们实现了以下的目标：

提高数据质量：原始数据中存在垃圾评论和不相关内容，通过数据预处理，清洗掉了这些无效数据，确保分析基于准确和相关的信息。

格式统一和标准化：原始数据集中的数据格式可能需要标准化，以便于分析。我们通过预处理，判断了数据格式都正确不需要修改。

提升分析效率：我们通过移除无关和冗余的字段，将 10 个字段的数据集缩减为了 5 个字段，极大地减少数据集的大小，从而提高了数据处理和分析的效率、减少了数据储存的空间成本。

便于后续分析：预处理后的数据便于进行各种统计和机器学习分析，例如情感分析、趋势分析等。

无重复：重复数据会对分析结果的准确性造成一定影响，预处理后的评论数据不存在重复数据。

3.数据分析

我们对于数据分析部分，进行了下面的分析，并得出了一系列的结果：

1.**垃圾评论描述性统计分析**：经过之前的数据预处理步骤，我们实现了垃圾评论的筛选，进一步的，我们可以探索垃圾评论的出现有何规律?电影评论平台可以基于这些规律来帮助识别和过滤垃圾评论，提升评论质量，以增强用户对平台的信任和满意度。

2.评论数量的时间趋势：分析电影评论数量随时间的变化，探索电影热度的动态变化。电影制片方和发行商可以利用时间趋势分析调整宣传策略，如在热度下降时增加广告投放，或者在特定时期推出特别活动以维持关注度。

3.平均评分分析：即分析各个电影的平均评分，或者各种情感倾向的评论、各种评论种类的平均评分。平均评分对电影评价网站尤为重要，它不仅帮助用户选择值得观看的电影，还对电影的长期声誉和收益产生影响。

4.点赞数量分析：内容创作者和市场营销人员可以依据点赞数来识别哪些评论或观点与观众产生共鸣，从而在未来的创作或营销中重点强调这些元素。

5.关键词分析：识别评论中的最受观众喜爱的角色和观众关注的电影元素。电影分析师和市场研究人员可以通过关键词来追踪市场趋势，理解观众的兴趣点，指导未来的电影制作和市场策略。例如：探究最受欢迎的角色，可以帮助电影发行商受喜爱的角色来设计海报封面，从而赢得更多好感。也可以帮助电影发行商的周边商品的选择，后续作品演员选择。探究最受欢迎的角色，可以帮助电影发行商在撰写海报以及推文标题的时候，可以加入这些元素关键词，从而达到吸引观众的效果。

6.情感倾向分析：即区分每条评论的积极、消极或中立倾向，了解观众的情感反应。电影制片方可以通过情感分析了解观众对电影的情感反应，以此指导电影剧本的修改，甚至影响后续作品的创作方向。

7.评论类别分析：即对每一条评论进行分类，以探索观众对电影不同方面（如剧情、特效）的关注。电影制片方、营销团队和影评人可以利用评论分类来深入理解观众对电影不同方面（如角色、剧情、特效）的看法，指导电影的改进和营销策略。

8.评论与评分的相关性：通过分析评论内容与电影评分之间的关系，可以理解评论对观众决策的影响，帮助电影评价平台优化评分算法。

9.电影特征与评论趋势的关联：电影市场营销团队可以通过分析不同电影特征（如类型、主演）与评论趋势的关系，来定位目标观众群体，优化宣传内容，提高市场竞争力。

4.数据挖掘

我们使用了三种数据挖掘方法，分别是：基于随机森林模型的评论情感倾向预测、基于 LDA 主题模型的评论分类预测、基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计。

基于随机森林模型的评论情感倾向预测：

1.粗略标记情感倾向。因为数据集中缺乏情感倾向的字段，而随机森林模型的训练需要大量的预先标记好情感倾向的评论数据，所以我们为了方便起见：1.将 4 星 5 星的评论归类为积极情感。2.将三星评论归类为中立情感。3.将 1 星 2 星归类为消极情感 。

2.然后我们对模型进行了一定的评价,可能是由于积极评论数量高出其他情绪的评论很多,所以对于积极评论的预测结果更准确,而其他情绪准确度依次下降。

3.然后，我们使用随机森林模型为每一条评论预测其情感倾向，我们还为数据集创建了一个新的字段，用来储存预测结果

基于 LDA 主题模型的评论分类预测：

1.数据集中没有每条评论主题，需要人工标注大量的评论来训练数据集，因此，为了方便起见，我们选择了无监督的 LAD 主题模型。

2.首先，我们从网上下载了停用词表，然后对评论进行分词，随后将评论文本转换成一个词频矩阵。

3.随后，我们训练 LDA 模型并使用。

4.然后，我们得出了第一次分类的结果，因为并不是很符合要求，所以我们又调整参数，进行了第二次主题建模。

5.接下来,我们使用 LAD 模型,对电影复仇者联盟的所有评论进行分类,使用训练好的 LDA 模型对文本数据集进行变换。从而获取每个主题在该评论中出现的概率。

6.通过查找每条评论的主题分布中概率最高的主题来确定其主要主题。

7.分类完成后,我们对各种类别进行了统计分析

基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计：

1.首先，我们通过创建角色映射表,即分别能够代表每个角色的关键词，来找到每个评论都提到了哪个角色。

2.由于电影《复仇者联盟》角色众多，且电影较为热门所以评论数据也多，于是我们以电影《复仇者联盟》为例进行了这一个文本挖掘方法。通过网上查找资料，我们得出了如图所示的角色映射表

3.生成反向映射表,我创建了一个名为 reverse_aliases 的字典，这是 character_aliases 的反转映射，用于将评论中的别名映射回标准角色名称。然后,我定义了一个名为 standardize_character_mentions 的函数，该函数接收一条评论作为输入，并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。最后,应用标准化函数,用于将评论中的别名映射回标准角色名称。并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。

4.对得出的结果进行可视化分析,以便于查看是否有误

采用的方法和技术

1. **八爪鱼采集器**:

使用模板采集和自定义采集方法从豆瓣网采集评论数据。

我们处理了八爪鱼采集器中采集数据量限制的问题。

2. **利用 Kaggle 网站进行数据集下载**:

利用 Kaggle 平台下载预处理过的数据集，作为分析的主要数据来源。

3. **Python**:

我们全程使用 Python 作为编程语言，用于执行数据处理和分析任务。

Python 提供了丰富的库和框架支持，简化了我们的模型构建过程。

4. **Jupyter Notebook**:

交互式编程环境，我们全程使用 Jupyter Notebook 来编写和执行 Python 代码。

Jupyter Notebook 支持代码、文本、图像和方程的混合展示，提高了分析过程的直观性和易懂程度。

5. **Scikit-learn (随机森林模型)**:

集成学习方法，构建多个决策树并合并结果以提高预测准确性。

易用性高，我们使用它进行情感倾向的预测分析。

6. **LDA 主题建模**:

无监督机器学习技术，用于从文本数据中识别不同的主题。

我们结合数据预处理步骤，如分词和去除停用词，用于分析文本数据并提取主题和关键词。

7. **自定义文本挖掘算法**:

创建角色和元素的映射表，通过文本分析提取评论中提及的角色和元素。

8. **Pyecharts**:

用于生成 ECharts 图表的 Python 库，是我们绘制图表的主要工具。

Pyecharts 可以创建三维散点图和三维柱状图，展示数据在多维度上的分布和趋势。

9. **Matplotlib 和 Seaborn**:

绘图工具，用于生成折线图、柱状图等。

我们使用 Matplotlib 和 Seaborn 作为次要绘图工具，用于快速绘图，以使结果更直观，快速分析结果。

10. **Pandas**:

数据处理和分析库，用于数据导入、清洗、处理和分析。

提供 DataFrame 对象，方便处理表格数据。

11. **NumPy**:

核心科学计算库，与 Pandas 结合使用，进行高效的数值计算。

12. **镝数图表**:

我们使用镝数图表绘制精美的可视化大屏、图文图片等。

美化图片和表格展示，提高分析报告的外观质量。

3、商务数据分析工具

与技术路线中的步骤和方法相对应，介绍使用什么软件或编程工具进行相关步骤的分析，并对采用各分析工具的原因进行简要说明。

数据采集采用的方法和技术

1. **八爪鱼采集器**:

我们选择八爪鱼采集器因为它提供了一个直观的方式来从网站上抓取数据。它的模板和自定义采集功能使我们能够高效地收集所需数据。

2. **Kaggle 数据集下载**:

Kaggle 平台被用于下载数据集，因为它是一个著名的数据科学社区，提供了大量预处理和高质量的数据集，节省了我们大量的数据准备时间。

另外，采集到的数据集数据量较小，只有 33000 条，一共 20 部电影，而网上下载的数据集则有 2125056 条数据，一共 28 部电影。

数据预处理采用的方法和技术

Python 和 Pandas 被用于数据预处理，因为它们可以有效地处理和转换数据集，保证数据质量和分析的准确性。

数据统计分析采用的方法和技术

1. **Pyecharts**:

Pyecharts 被选中因为它可以创建互动式且美观的图表，这对于呈现数据和洞见非常重要。

2. **Matplotlib 和 Seaborn**:

这两个库被用于数据可视化，因为它们提供了广泛的图表类型和高度定制的选项，能够清晰地传达复杂的数据故事。

3. **Pandas**:

Pandas 是数据处理和分析的首选库，因为它提供了强大的数据结构和功能，使得数据清洗和转换变得简单高效。

4. **NumPy**:

NumPy 用于处理大型多维数组和矩阵，非常适合进行高效的数值计算，这对于数据分析至关重要。

5. **镝数图表**:

镝数图表用于创建更加精美和专业的数据可视化，提升了我们报告和 PPT 的视觉吸引力。

数据挖掘采用的方法和技术

1. **Python**:

选择 Python 的原因在于它的强大和灵活性。它是数据科学领域广泛使用的语言，具有丰富的库和框架，能够简化复杂的数据处理和分析任务。

2. **Jupyter Notebook**:

使用 Jupyter Notebook 的主要原因是它的交互性和易用性。它允许我们在一个平台上编写代码、分析数据，并直观地展示结果，提高了工作效率。

3. **Scikit-learn (随机森林模型)**:

我们选择随机森林模型，因为它是一个强大的集成学习方法，能提供高准确性的预测。它适合于复杂的数据集，并且 Scikit-learn 的实现简单易用。

4. **LDA 主题建模**:

LDA 是一种有效的无监督学习技术，用于从文本中提取主题。我们选择它因为它可以帮助我们理解大量未标记文本数据的潜在主题结构。

5. **自定义文本挖掘算法**:

我们开发了自定义文本挖掘算法来提取特定信息（如角色提及），因为这种方法可以根据我们的具体需求定制，从而提供更精准的分析结果。

4、商务数据分析过程

4.1 数据描述

八爪鱼采集器采集到的数据:

大多数电影采集到的都是 1800 条评论数据，少部分电影因为某类评论过少或者总评论数量过少，导致采集到的评论少于 1800 条。因此，总评论数据大约有 33000 条。数据的储存方式为每部电影存放在一个 csv 文件下，一共 20 部电影，共 20 个 csv 类型的文件：

名称	类型	大小
奥本海默	Microsoft Excel 工作表	775 KB
第八个嫌疑人	Microsoft Excel 工作表	582 KB
毒液-致命守护者	Microsoft Excel 工作表	492 KB
亨利·休格的神奇故事	Microsoft Excel 工作表	344 KB
铃芽之旅	Microsoft Excel 工作表	732 KB
流浪地球	Microsoft Excel 工作表	742 KB
美人鱼	Microsoft Excel 工作表	460 KB
倩女幽魂	Microsoft Excel 工作表	433 KB
让子弹飞	Microsoft Excel 工作表	388 KB
三傻大闹宝莱坞	Microsoft Excel 工作表	374 KB
生化危机	Microsoft Excel 工作表	337 KB
泰坦尼克号	Microsoft Excel 工作表	504 KB
唐山大地震	Microsoft Excel 工作表	402 KB
无价之宝	Microsoft Excel 工作表	378 KB
星际穿越	Microsoft Excel 工作表	433 KB
寻梦环游记	Microsoft Excel 工作表	499 KB
长安三万里	Microsoft Excel 工作表	764 KB
长城	Microsoft Excel 工作表	514 KB
珍·古道尔的传奇一生	Microsoft Excel 工作表	357 KB
拯救嫌疑人	Microsoft Excel 工作表	559 KB
志愿军-雄兵出击	Microsoft Excel 工作表	552 KB

	点赞数	时间	评论内容	星级
0	2986	\n 2023-09-28 20:22:31\n ...	建议后两部直接上完整体。	
1	2217	\n 2023-09-28 23:20:43\n ...	总体不错，少有的把为什么一定要打这场仗讲清楚的，不过因为讲清楚了所以前半部分都是指南，后面节...	
2	1162	\n 2023-09-28 20:27:38\n ...	黄土地、霸王别姬、喜的是陈凯歌拍的吗？	
3	2067	\n 2023-09-28 21:16:04\n ...	看完超出预期，展现有限的知识储备，没看出夹带私货。本来就是全景式展现抗美援朝战争的片子，评论...	
4	602	\n 2023-10-03 14:15:42\n ...	如坐针毡，如芒刺背，节奏乱的一批，凯哥用一部部作品证明了《霸王别姬》不是他拍的。	
...	...	...	...	...
1795	3	\n 2023-10-02 17:55:33\n ...	矫情！ 没体现出政治智慧，喷才大都	
1796	1	\n 2023-09-29 12:33:25\n ...	摄影、剪辑以后都可以别提括儿了，前脚刚被敌死队4的低工业水准和成片质量震撼，后脚就又来一部极...	
1797	0	\n 2023-10-01 21:00:21\n ...	张飞宇确实是个阳光开朗大男孩 我是陈凯歌也会很喜欢这个儿子	
1798	4	\n 2023-09-30 14:47:20\n ...	差就是差，别和我扯什么爱国，什么抗意。老陈啊，为什么要承为自己做不擅长的事呢？你和老张都不适...	
1799	1	\n 2023-10-02 09:49:20\n ...	1.6四舍五入 多么有意义的题材拍成了爱国网大加电视剧，我以为2023年的电影里不会有样板戏...	

1800 rows × 4 columns

一共有：点赞数、时间、评论内容、星级这几个字段，都是我们所需要的，另外我们还需要电影名的字段，这个在后续很容易进行增添，另外，时间和星级字段的内容需要进行处理，这也非常容易。

网上下载的数据:

网上下载的数据集的信息如下所示，一共 2125056 条评论数据，一共 28 部涵盖了各种类别的电影。包含了我们所需要的全部字段，数据集的储存方式为储存在同一个 csv 文件中，一共 406MB：

	ID	Movie_Name_EN	Movie_Name_CN	Crawl_Date	Number	Username	Date	Star	Comment	Like
0	0	Avengers Age of Ultron	复仇者联盟2	2017-01-22	1	然潘	2015-05-13	3	连美创都知道整容要去韩国。	2404
1	1	Avengers Age of Ultron	复仇者联盟2	2017-01-22	2	更深的白色	2015-04-24	2	非常失望，剧本完全敷衍了事，主线剧情没突破大家可以理解，可所有的人物都缺乏动机，正邪之间，...	1231
2	2	Avengers Age of Ultron	复仇者联盟2	2017-01-22	3	有意识的贱民	2015-04-26	2	2015年度最失望作品。以为面面俱到，实则画蛇添足；以为主题深刻，实则老调重弹；以为推陈出...	1052
3	3	Avengers Age of Ultron	复仇者联盟2	2017-01-22	4	不老的李大爷眼	2015-04-23	4	《钢铁人2》中勾引钢铁侠，《妇联1》中勾引黑寡妇，《美队2》中勾引美国队长，在《妇联2》中终于...	1045
4	4	Avengers Age of Ultron	复仇者联盟2	2017-01-22	5	ZephyrO	2015-04-22	2	虽然从头打到尾，但是真的很无聊啊。	723
...	...	...	...	...	...	...	...	...	...	...
2125051	2125051	Zootopia	疯狂动物城	2017-01-04	141196	猫的喷火尾巴	2016-03-06	4	真好看 兔子警官又单告善良又可爱~简直理想结婚对象！每一个动物造型都是那么赞~	0
2125052	2125052	Zootopia	疯狂动物城	2017-01-04	141197	Tosta	2016-03-05	5	六星好评！像头脑特工队那样惊喜！	0
2125053	2125053	Zootopia	疯狂动物城	2017-01-04	141198	風立ちぬ	2016-03-11	4	欢乐而又深刻，是童话故事更是政治寓言。	0
2125054	2125054	Zootopia	疯狂动物城	2017-01-04	141199	PITT	2016-03-05	5	对现实世界歧视和偏见的影射妙哉妙哉，不要害怕打破常规，try everything	0
2125055	2125055	Zootopia	疯狂动物城	2017-01-04	141200	普赖尔精斯	2016-03-06	5	萌物包装的政治正确片，那个叫Doug穿黄衫戴防毒面具的绵羊的处于对Walter和Jessi...	0

2125056 rows × 10 columns

预处理后的数据:

数据预处理后，数据的条数为 2118316 条，储存形式为在 python 代码的变量中，后续也可以导出为 csv 文件。数据的形式为：

ovie_Name_Cl	Date	Star	Comment	Like
复仇者联盟	2012-04-28	5	那些个说是大乱炖、	1297
复仇者联盟	2012-04-27	5	从头燃到尾！各种爽	1216
复仇者联盟	2012-04-23	2	纯粹狗血片，里面咋	771
复仇者联盟	2012-04-26	5	Hulk...Smash！！！	763
复仇者联盟	2012-04-27	2	看得想睡觉，3d怎么	652
复仇者联盟	2012-04-25	5	射了！！！！射了我	630
复仇者联盟	2012-04-23	4	这些漫威怪物集中起	591
复仇者联盟	2012-04-23	5	真没想到《复仇者联	569
复仇者联盟	2012-04-23	4	欢乐养眼！唐尼吐槽	352
复仇者联盟	2012-05-05	5	撸妇联前：诶嘿嘿好	331
复仇者联盟	2012-05-05	4	一部几乎完美的爆米	312
复仇者联盟	2012-05-05	5	IMAX3D打斗效果从与	288
复仇者联盟	2012-05-06	5	《复仇者联盟》国内	360
复仇者联盟	2012-05-05	3	被严重高估的粉丝电	269
复仇者联盟	2012-04-26	5	Loki真心美貌;Robi	247
复仇者联盟	2012-04-23	5	作为超级英雄和mary	188
复仇者联盟	2012-04-24	5	碉堡了！打绝人寰！	165
复仇者联盟	2012-05-07	1	被豆瓣评分欺骗，简	146
复仇者联盟	2012-05-06	5	我虐我哥千百遍 我哥	157
复仇者联盟	2012-04-26	5	还是钢铁侠最帅，和	78

可以看到，数据预处理后，只剩下了电影中文名称、评论日期、评分、评论内容、点赞数这五个字段，而原来的数据集中则含有 10 个字段。

4.2 数据分析过程

在数据分析中，我们主要分为数据统计分析和数据挖掘分析。这两种分析相辅相成，数据统计分析为数据挖掘分析提供了一个数据的大致概览，使得数据挖掘分析的进行更加游刃有余；数据挖掘分析使得能够挖掘每一条评论的情感倾向、分类、提到的角色和元素，为数据统计分析提供了更多的角度，丰富了数据统计分析。因此，在我们的项目中，我们的数据挖掘分析和数据统计分析是相辅相成，穿插进行的。

下面分别介绍数据统计分析和数据挖掘分析的过程：

数据统计分析（以绘制各个电影的垃圾评论平均评分、所有评论的平均评分三维柱状图为例）

1.导入所需库和数据：

我们首先导入了 Pandas 库，因为它是 Python 中用于数据处理和分析的主要库之一。

然后，我们加载了两个 CSV 文件：'结合白名单后筛选出的垃圾评论_阈值设定为1.csv' 和 'DMSC.csv'。第一个文件包含垃圾评论的数据，而第二个文件包含所有的电影评论数据。

2.计算垃圾评论的平均评分：

使用 Pandas 的 groupby 方法，我们按电影名称对垃圾评论数据进行了分组，并计算了每部电影的垃圾评论平均评分。

然后，我们使用 reset_index 方法将结果转换成一个新的 DataFrame，方便后续的合并操作。

3.计算所有评论的平均评分：

类似地，我们对所有评论数据执行了相同的操作，计算了每部电影的所有评论的平均评分，并将结果转换成另一个 DataFrame。

4.合并两个数据集：

使用 Pandas 的 merge 方法，我们将垃圾评论的平均评分和所有评论的平均评分合并成一个新的 DataFrame。这里我们按照电影名称（'Movie_Name_CN'）作为连接键，采用内连接的方式进行合并。

合并后，我们重命名了合并 DataFrame 的列名称为 'Movie_Name_CN'（电影名称）、'Spam_Avg_Rating'（垃圾评论平均评分）和 'All_Avg_Rating'（所有评论平均评分）。

5.查看合并结果：

最后，我们使用 head 方法查看了合并后 DataFrame 的前几行数据，以确保数据合并正确无误。

6.然后，我们使用这个数据集，使用 pyecharts 来绘制图形。具体操作如下：

数据准备：

首先，我们创建了一个空列表 data，这将用来存储绘制图表所需的数据。

接着，我们遍历 merged_ratings 这个 DataFrame，将每部电影的垃圾评论平均评分和所有评论的平均评分加入到 data 列表中。

对于每部电影，我们的数据是一个包含三个元素的小列表：电影索引、评论类型（0 代表垃圾评论，1 代表所有评论）、平均评分。

创建 3D 柱状图：

我们使用 Pyecharts 的 Bar3D 创建了一个三维柱状图对象。

在这个图中，我们添加了数据，并设置了 X 轴为电影种类、Y 轴为评论类型（垃圾评论和所有评论）、Z 轴为平均评分。

设置全局配置：

我们为图表添加了标题，并设置了视觉映射参数，其中最高分设置为 5 分，最低分设置为 1.6 分，以适应评分的实际范围。

图表渲染：

最后，我们将这个三维柱状图渲染成一个 HTML 文件，命名为“电影名垃圾评论平均评分所有评论平均评分三维.html”，便于后续的展示和分析。

数据挖掘分析

数据挖掘分析中，我将分别介绍基于随机森林模型的评论情感倾向预测、基于 LDA 主题模型的评论分类预测、基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计，这三种数据挖掘算法的具体步骤。

基于随机森林模型的评论情感倾向预测：

1.粗略标记情感倾向。因为数据集中缺乏情感倾向的字段，而随机森林模型的训练需要大量的预先标记好情感倾向的评论数据，所以我们为了方便起见：1.将 4 星 5 星的评论归类为积极情感。2.将三星评论归类为中立情感。3.将 1 星 2 星归类为消极情感 。

2.然后我们对模型进行了一定的评价,可能是由于积极评论数量高出其他情绪的评论很多,所以对于积极评论的预测结果更准确,而其他情绪准确度依次下降。

3.然后，我们使用随机森林模型为每一条评论预测其情感倾向，我们还为数据集创建了一个新的字段，用来储存预测结果

基于 LDA 主题模型的评论分类预测：

1.数据集中没有每条评论主题，需要人工标注大量的评论来训练数据集，因此，为了方便起见，我们选择了无监督的 LAD 主题模型。

2.首先，我们从网上下载了停用词表，然后对评论进行分词，随后将评论文本转换成一个词频矩阵。

3.随后，我们训练 LDA 模型并使用。

4.然后，我们得出了第一次分类的结果，因为并不是很符合要求，所以我们又调整参数，进行了第二次主题建模。

5.接下来,我们使用 LAD 模型,对电影复仇者联盟的所有评论进行分类,使用训练好的 LDA 模型对文本数据集进行变换。从而获取每个主题在该评论中出现的概率。

6.通过查找每条评论的主题分布中概率最高的主题来确定其主要主题。

7.分类完成后,我们对各种类别进行了统计分析

基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计：

1.首先，我们通过创建角色映射表,即分别能够代表每个角色的关键词，来找到每个评论都提到了哪个角色。

2.由于电影《复仇者联盟》角色众多，且电影较为热门所以评论数据也多，于是我们以电影《复仇者联盟》为例进行了这一个文本挖掘方法。通过网上查找资料，我们得出了如图所示的角色映射表

3.生成反向映射表,我创建了一个名为 reverse_aliases 的字典，这是 character_aliases 的反转映射，用于将评论中的别名映射回标准角色名称。然后,我定义了一个名为 standardize_character_mentions 的函数，该函数接收一条评论作为输入，并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。最后,应用标准化函数,用于将评论中的别名映射回标准角色名称。并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。

4.对得出的结果进行可视化分析,以便于查看是否有误

4.3 结果分析

详细说明数据采集、数据预处理、数据统计分析及数据挖掘分析的过程及结果；对于数据统计分析及数据挖掘方法，需要介绍所采用的模型或算法、建模过程、算法参数设置等；附必要的软件运行界面截图、关键程序代码、结果截图或列表等。

1.数据采集的过程及结果

1. **八爪鱼采集器的使用**：

我负责使用八爪鱼数据采集器采集数据，这是我第一次使用这样的工具，因此刚开始有些不熟悉。

通过在 Bilibili 上查找相关教程，我学会了如何操作八爪鱼采集器。

我尝试了两种方法：模板采集法和自定义采集法。在模板采集法中，虽然采集过程快速，但数据质量不理想。因此，我转而尝试了自定义采集法，这需要更多的手动设置，但能够更精确地控制采集的数据。

在模板采集法中，我们一共找到了两个合适的模板。在使用第一个模板时，我们遇到采集效果不理想的问题，采集到的是追评，即采集的是对于评论的回复，而非初始评论。

经过分析，发现，因为豆瓣网中的每个评论都可以有追评，所以这个模板把所有的追评（及评论下方的回复）都爬取了下来，结合实际，评论的追评部分往往是网友之间互相争吵的地方，因此，这些追评并不是我们数据爬取想要的结果，而真实的、被回复的、原来的评论则不断重复，很多条数据中才会出现一条这样的评论，因此，不能使用这个模板。于是我们开始尝试第二个模板。

在使用第二个模板时，我们遇到了采集数量上限的问题，只能采集前 220 条的评论数据。然后，我们检查了八爪鱼采集器的各种设置，并查找了有关的问题，但是没有找到相关的结果，对八爪鱼的配置进行了修改也没有成功突破数据采集的数量限制。另外，这个模板采集速度过慢，因此，我们尝试使用其他方法，即自定义采集。

在自定义采集中，我学会了设置循环翻页、选取评论列表，并提取相关字段。我还发现了采集数据量的上限问题，仅能采集到前 220 条评论数据。

2. **数据采集的限制和解决方法**：

我们在采集过程中遇到了数据量限制的问题。我们似乎每个电影都只能收集前 220 条数据。为了解决这个问题，我们尝试了多种方法，排查了各种原因，可能的原因有：到底是因为豆瓣只提供了前 220 条数据可供查看？还是因为八爪鱼采集器的限制？又或者是豆瓣网上有着一定的反爬机制造成的限制？

带着这些问题，回到了豆瓣网的《长城》的评论区，如下图所示，我们选择的是看过的人的全部的评论，所以应该能够显示出所有的评论才对：

然后，我们检查了八爪鱼采集器的各种设置，并查找了有关的问题，但是没有找到相关的结果，对八爪鱼的配置进行了修改也没有成功突破数据采集的数量限制：

最终，我们发现这是因为我们没有登录账号，所以只能看到前 11 页的数据，而前 11 页的数据刚好 220 条，这恰恰说明了我们是由于没有登录账号导致采集数量受限。

随后我们登录账号，但是即使登录，也只能查看前 30 页的评论数据，一共 600 条，我们尝试解决但是无果。

我们为了扩充数据量，决定分别从好评、一般、差评中获取不重复的评论数据。这样，每部电影就可以采集到前 1800 条数据。

3. **从网上下载数据集**：

通过网上查找数据集获取的方法，我们了解到 Kaggle 作为一个知名的数据科学社区，常被数据分析师和研究人员用来分享和寻找数据集。

Kaggle 是一个著名的数据科学竞赛平台，提供了广泛的数据集资源。在获取数据集的步骤中，我们首先登录 Kaggle 网站，然后导航至 Datasets 部分。在这里，我们利用搜索功能查找到所需的电影评论数据集。

找到目标数据集后，我们进一步浏览了数据集的描述、数据量、字段信息等，以确保数据集符合我们的分析需求。确认无误后，我们下载了数据集。下载的数据集以 CSV 文件格式存储，这也便于我们后续在 Python 等数据分析工具中进行操作和分析。

4. **数据集的选择**（最终选择只使用 Kaggle 上面下载的数据集）：

由于两种方式获得的数据集有很大的差异，所以我们不能够简单的将两个数据集相整合来使用。具体原因如下：因为他们包含大量重复的数据集，而且选取的电影也基本不相同，而且从八爪鱼采集器采集到的 2017 年以后的评论数据也会因为数据量过少被稀释掉，从而难以发挥作用，同时也会导致不同电影之间，评论的数量有着很大的差异。

两种数据集的具体对比分析如下：

数据集	优点	缺点
八爪鱼采集器采集到的数据集	评论内容是最新的，电影内容也是最新的，根据豆瓣电影的分类，每一种各选了一部电影，同时也包含了中国，美国，英国等地的电影，含有我们所需的所有字段，数据预处理简单。	数据量相对过少。

从网站 Kaggle 上下载的数据集	数据量十分庞大，含有我们所需的所有字段，数据预处理简单。	评论内容和电影并不是最新的，因为爬取的时间是在 2017 年，因此数据集中只有这之前的评论和电影数据。

因此两种数据集各有优缺点

结合我们的研究目的，我们主要是需要根据评论数据来获取对于电影发行商，消费者，影评网站的建议，这需要庞大的数据量的支持，因此，从网站 Kaggle 上下载的数据集作为我们的研究使用的数据集更加合适。因此，在后续研究中，我们都使用从网站 Kaggle 上下载的数据集作为研究对象。

2.数据预处理的过程及结果

1.原始数据集的分析

原始数据集的信息如下所示，一共 2125056 条评论数据，一共 28 部涵盖了各种类别的电影。包含了我们所需要的全部字段，数据集的储存方式为储存在同一个 csv 文件中，一共 406MB：

我们发现，数据集中包含了很多我们不需要的字段，需要删除，

其次，日期字段的格式需要确保是能够被 python 处理的格式。星级，点赞数，评论内容，电影中文名这几个字段分别都以正确、易处理的的形式所储存（整型、字符串），因此不需要进行数据类型的转化。

另外，通过观察发现，评论数据中有一些和电影无关的内容（广告,无关评论等），我们将其称作垃圾评论,这些无关评论对数据分析的准确性造成了一定影响，因此需要对这些垃圾评论进行筛选并删除。

2.分析数据预处理的步骤

在我们的项目中，数据预处理的关键步骤包括筛选重要字段、删除重复值和缺失值、以及过滤垃圾评论。通过这些步骤，我们旨在提高数据的质量和分析的准确性，以减少数据集的大小和维度。例如，筛选重要字段是为了减少数据集的大小和维度，节省后续操作的运行时间和空间。删除重复值和缺失值是为了确保数据的完整性和代表性；而垃圾评论的过滤则是为了提高数据分析的有效性和准确性。

3.删除重复值

通过分析，我们发现数据集中存在 16 条重复评论，因此删除了其中的 8 条评论

4.删除缺失值

我们经过分析，发现数据集中不存在缺失值

5.筛选重要字段

为了减少数据集的大小和维度，加快后续操作的运行，同时节省储存空间。我们使用 python 筛选出了最重要的五个字段，即：电影名、评论日期、评论内容、星级评分、点赞数

6.统一字段的数据格式

经过分析，我们发现我们的筛选后的五个字段的数据格式都是统一的，评论日期为日期形式，电影名和评论内容为文本型，星级评分和点赞数为整数型。

7.过滤垃圾评论

- 我们的第一步是设置黑名单,并设定一个阈值,如果一个评论满足阈值数量的黑名单中的关键词，则被标记为垃圾评论
- 第二步是设置白名单,通过设置白名单，对所有第一次标记为垃圾评论的评论进行纠正，重新标记为正常评论
- 第三步是经过黑白名单得出垃圾评论结果,并查看效果

- 然后,我们经过不断改变黑白名单中的关键词,并查看效果,观察哪些黑名单中的关键词设置的不好，总是误将正常评论标记为垃圾评论，就将该关键词删除,或者继续增添白名单中的关键词，修改误标记的评论。
- 然后我们查看了阈值分别设定为 1 和 2，并不断更改关键词之后，得到的结果：可以看到阈值设定为 2 时，真正垃圾评论的占比有显著提升.但是经过该方法,筛选到的垃圾评论总量只有 406 条,而将阈值设定为 1 时,筛选到的总评论数量总有 6732 条.因此两种方法各有优劣,阈值设定为 2 筛选出的垃圾评论非常准确,但是可能会遗漏很多垃圾评论,使他们没有被筛选出来。最终,我们采取了将阈值设定为 1 的方法，过滤掉了数据集中存在的垃圾评论。
- 然后，我们分析了黑白名单的关键词的作用性大小。
- 其中，我们将黑名单关键词的作用定义为在垃圾评论中,该关键词出现的次数。白名单关键词的作用定义为,在黑名单初步筛选得出的结果中,白名单关键词出现的次数。

经过数据预处理，数据的条数为 2118316 条，储存形式为在 python 代码的变量中，我们实现了以下的目标：

提高数据质量：原始数据中存在垃圾评论和不相关内容，通过数据预处理，清洗掉了这些无效数据，确保分析基于准确和相关的信息。

格式统一和标准化：原始数据集中的数据格式可能需要标准化，以便于分析。我们通过预处理，判断了数据格式都正确不需要修改。

提升分析效率：我们通过移除无关和冗余的字段，将 10 个字段的数据集缩减为了 5 个字段，极大地减少数据集的大小，从而提高了数据处理和分析的效率、减少了数据储存的空间成本。

便于后续分析：预处理后的数据便于进行各种统计和机器学习分析，例如情感分析、趋势分析等。

无重复：重复数据会对分析结果的准确性造成一定影响，预处理后的评论数据不存在重复数据。

3.数据分析的过程及结果

我们对于数据分析部分，进行了下面的分析，并得出了一系列的结果：

1.**垃圾评论描述性统计分析**：经过之前的数据预处理步骤，我们实现了垃圾评论的筛选，进一步的，我们可以探索垃圾评论的出现有何规律?电影评论平台可以基于这些规律来帮助识别和过滤垃圾评论，提升评论质量，以增强用户对平台的信任和满意度。

2.评论数量的时间趋势：分析电影评论数量随时间的变化，探索电影热度的动态变化。电影制片方和发行商可以利用时间趋势分析调整宣传策略，如在热度下降时增加广告投放，或者在特定时期推出特别活动以维持关注度。

3.平均评分分析：即分析各个电影的平均评分，或者各种情感倾向的评论、各种评论种类的平均评分。平均评分对电影评价网站尤为重要，它不仅帮助用户选择值得观看的电影，还对电影的长期声誉和收益产生影响。

4.点赞数量分析：内容创作者和市场营销人员可以依据点赞数来识别哪些评论或观点与观众产生共鸣，从而在未来的创作或营销中重点强调这些元素。

5.关键词分析：识别评论中的最受观众喜爱的角色和观众关注的电影元素。电影分析师和市场研究人员可以通过关键词来追踪市场趋势，理解观众的兴趣点，指导未来的电影制作和市场策略。例如：探究最受欢迎的角色，可以帮助电影发行商受喜爱的角色来设计海报封面，从而赢得更多好感。也可以帮助电影发行商的周边商品的选择，后续作品演员选择。探究最受欢迎的角色，可以帮助电影发行商在撰写海报以及推文标题的时候，可以加入这些元素关键词，从而达到吸引观众的效果。

6.情感倾向分析：即区分每条评论的积极、消极或中立倾向，了解观众的情感反应。电影制片方可以通过情感分析了解观众对电影的情感反应，以此指导电影剧本的修改，甚至影响后续作品的创作方向。

7.评论类别分析：即对每一条评论进行分类，以探索观众对电影不同方面（如剧情、特效）的关注。电影制片方、营销团队和影评人可以利用评论分类来深入理解观众对电影不同方面（如角色、剧情、特效）的看法，指导电影的改进和营销策略。

8.评论与评分的相关性：通过分析评论内容与电影评分之间的关系，可以理解评论对观众决策的影响，帮助电影评价平台优化评分算法。

9.电影特征与评论趋势的关联：电影市场营销团队可以通过分析不同电影特征（如类型、主演）与评论趋势的关系，来定位目标观众群体，优化宣传内容，提高市场竞争力。

4.数据挖掘的过程及结果

我们使用了三种数据挖掘方法，分别是：基于随机森林模型的评论情感倾向预测、基于 LDA 主题模型的评论分类预测、基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计。

基于随机森林模型的评论情感倾向预测：

1.粗略标记情感倾向。因为数据集中缺乏情感倾向的字段，而随机森林模型的训练需要大量的预先标记好情感倾向的评论数据，所以我们为了方便起见：1.将 4 星 5 星的评论归类为积极情感。2.将三星评论归类为中立情感。3.将 1 星 2 星归类为消极情感 。

2.然后我们对模型进行了一定的评价,可能是由于积极评论数量高出其他情绪的评论很多,所以对于积极评论的预测结果更准确,而其他情绪准确度依次下降。

3.然后，我们使用随机森林模型为每一条评论预测其情感倾向，我们还为数据集创建了一个新的字段，用来储存预测结果

基于 LDA 主题模型的评论分类预测：

1.数据集中没有每条评论主题，需要人工标注大量的评论来训练数据集，因此，为了方便起见，我们选择了无监督的 LAD 主题模型。

2.首先，我们从网上下载了停用词表，然后对评论进行分词，随后将评论文本转换成一个词频矩阵。

3.随后，我们训练 LDA 模型并使用。

4.然后，我们得出了第一次分类的结果，因为并不是很符合要求，所以我们又调整参数，进行了第二次主题建模。

5.接下来,我们使用 LAD 模型,对电影复仇者联盟的所有评论进行分类,使用训练好的 LDA 模型对文本数据集进行变换。从而获取每个主题在该评论中出现的概率。

6.通过查找每条评论的主题分布中概率最高的主题来确定其主要主题。

7.分类完成后,我们对各种类别进行了统计分析

基于自定义文本挖掘算法的受欢迎的角色和元素倾向统计：

1.首先，我们通过创建角色映射表,即分别能够代表每个角色的关键词，来找到每个评论都提到了哪个角色。

2.由于电影《复仇者联盟》角色众多，且电影较为热门所以评论数据也多，于是我们以电影《复仇者联盟》为例进行了这一个文本挖掘方法。通过网上查找资料，我们得出了如图所示的角色映射表

3.生成反向映射表,我创建了一个名为 reverse_aliases 的字典，这是 character_aliases 的反转映射，用于将评论中的别名映射回标准角色名称。然后,我定义了一个名为 standardize_character_mentions 的函数，该函数接收一条评论作为输入，并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。最后,应用标准化函数,用于将评论中的别名映射回标准角色名称。并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。

4.对得出的结果进行可视化分析,以便于查看是否有误

5.数据统计分析采用的方法、必要的软件运行界面截图、关键程序代码、结果截图或列表等。

数据统计分析采用的方法

（以绘制各个电影的垃圾评论平均评分、所有评论的平均评分三维柱状图为例，由于在编写代码时，代码的先后顺序比较混乱，而且有可能覆盖了之前写的代码和结果，所以无法全部找出数据统计分析的代码和其对应的结果，所有我们再次省略了其他图表的具体实现代码，但是在结果部分会展示全部的结果）

1.导入所需库和数据：

我们首先导入了 Pandas 库，因为它是 Python 中用于数据处理和分析的主要库之一。

然后，我们加载了两个 CSV 文件：'结合白名单后筛选出的垃圾评论_阈值设定为1.csv' 和 'DMSC.csv'。第一个文件包含垃圾评论的数据，而第二个文件包含所有的电影评论数据。

2.计算垃圾评论的平均评分：

使用 Pandas 的 groupby 方法，我们按电影名称对垃圾评论数据进行了分组，并计算了每部电影的垃圾评论平均评分。

然后，我们使用 reset_index 方法将结果转换成一个新的 DataFrame，方便后续的合并操作。

3.计算所有评论的平均评分：

类似地，我们对所有评论数据执行了相同的操作，计算了每部电影的所有评论的平均评分，并将结果转换成另一个 DataFrame。

4.合并两个数据集：

使用 Pandas 的 merge 方法，我们将垃圾评论的平均评分和所有评论的平均评分合并成一个新的 DataFrame。这里我们按照电影名称（'Movie_Name_CN'）作为连接键，采用内连接的方式进行合并。

合并后，我们重命名了合并 DataFrame 的列名称为 'Movie_Name_CN'（电影名称）、'Spam_Avg_Rating'（垃圾评论平均评分）和 'All_Avg_Rating'（所有评论平均评分）。

5.查看合并结果：

最后，我们使用 head 方法查看了合并后 DataFrame 的前几行数据，以确保数据合并正确无误。

6.然后，我们使用这个数据集，使用 pyecharts 来绘制图形。具体操作如下：

7.数据准备：

首先，我们创建了一个空列表 data，这将用来存储绘制图表所需的数据。

接着，我们遍历 merged_ratings 这个 DataFrame，将每部电影的垃圾评论平均评分和所有评论的平均评分加入到 data 列表中。

对于每部电影，我们的数据是一个包含三个元素的小列表：电影索引、评论类型（0 代表垃圾评论，1 代表所有评论）、平均评分。

8.创建 3D 柱状图：

我们使用 Pyecharts 的 Bar3D 创建了一个三维柱状图对象。

在这个图中，我们添加了数据，并设置了 X 轴为电影种类、Y 轴为评论类型（垃圾评论和所有评论）、Z 轴为平均评分。

9.设置全局配置：

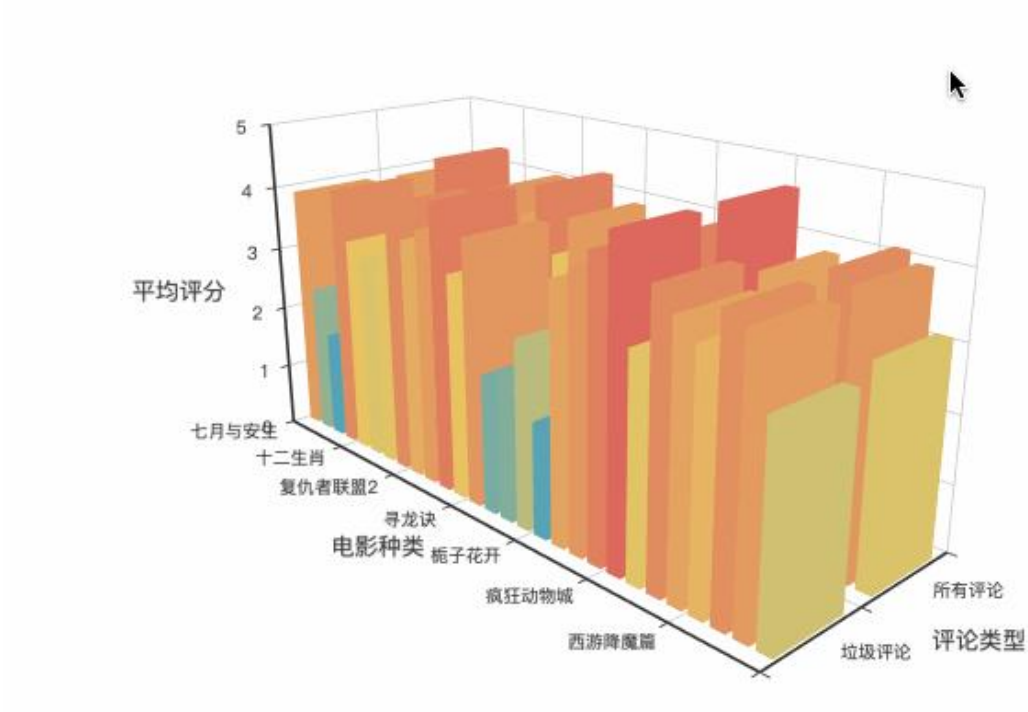
我们为图表添加了标题，并设置了视觉映射参数，其中最高分设置为 5 分，最低分设置为 1.6 分，以适应评分的实际范围。

10.图表渲染：

最后，我们将这个三维柱状图渲染成一个 HTML 文件，命名为“电影名垃圾评论平均评分所有评论平均评分三维.html”，便于后续的展示和分析。

结果展示：

（双击可全屏播放动图，然后按 Esc 退出）



6.数据挖掘采用的模型或算法、建模过程、算法参数设置等以及必要的软件运行界面截图、关键程序代码、结果截图或列表等。

一、基于随机森林模型的评论情感倾向预测

粗略标记情感倾向

因为数据集中缺乏情感倾向的字段，而随机森林模型的训练需要大量的预先标记好情感倾向的评论数据，所以我们为了方便起见：

1.将 4 星 5 星的评论归类为积极情感

2.将三星评论归类为中立情感

3.将 1 星 2 星归类为消极情感

```
#粗略标记情感倾向
def label_sentiment(row):
    if row['Star'] >= 4:
        return 'positive'
    elif row['Star'] == 3:
        return 'neutral'
    else:
        return 'negative'

df1['Sentiment'] = df1.apply(label_sentiment, axis=1)
```

然后我们查看结果，发现标记成功：

```
df1.head()
```

	Movie_Name_CN	Number	Username	Date	Star	Comment	Like	Topic_Label	Sentiment
1073583	复仇者联盟	1	花生狼	2012-04-28	5	那些个说是大乱炖、狗血片的人，拜托你们闭嘴吧，你不看marvel漫画不怪你，但什么都不知道...	1297	感受	positive
1073584	复仇者联盟	2	LORENZO	2012-04-27	5	从头燃到尾！各种爽！各种给力！唐尼大眼睛各种吐槽！浩克裸一把还卖萌，你是天神我膜拜！锤哥一...	1216	角色	positive
1073585	复仇者联盟	3	麦子	2012-04-23	2	纯粹狗血片，里面咋没有孙悟空、金刚葫芦娃和哪吒！	771	特征	negative
1073586	复仇者联盟	4	陀螺凡达可	2012-04-26	5	Hulk...Smash！！！	763	期待	positive
1073587	复仇者联盟	5	五色全味	2012-04-27	2	看得想睡觉，3d怎么还不死啊，纽约要被拆多少遍	652	特效	negative

然后，我们创建随机森林模型，并利用这些数据，训练随机森林模型：

```
import jieba
import re

def preprocess(text):
    text = re.sub(r'[^\w\s]', '', text)  # 去除标点符号
    words = jieba.cut(text)  # 分词
    return ' '.join(words)

df1['Processed_Comment'] = df1['Comment'].apply(preprocess)
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_df=0.9, min_df=2)
X = vectorizer.fit_transform(df1['Processed_Comment'])
def label_sentiment(row):
    if row['Star'] >= 4:
        return 'positive'
    elif row['Star'] == 3:
        return 'neutral'
    else:
        return 'negative'

df1['Sentiment'] = df1.apply(label_sentiment, axis=1)
from sklearn.model_selection import train_test_split

y = df1['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)
from sklearn.metrics import classification_report

y_pred = classifier.predict(X_test)
print(classification_report(y_test, y_pred))
```

然后，我们输出模型的预测结果，查看预测准确度：

模型评价

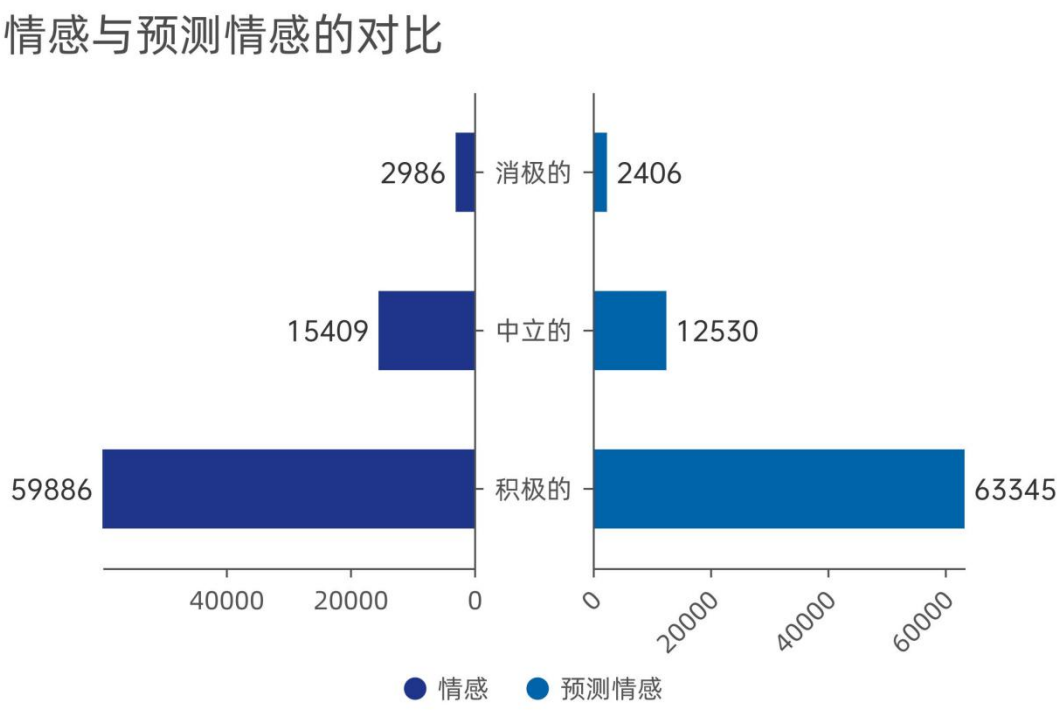
	precision	recall	f1-score	support
消极	0.35	0.09	0.15	570
中立	0.47	0.16	0.24	3083
积极	0.8	0.96	0.87	12004

模型结果如上图所示,可能是由于积极评论数量高出其他情绪的评论很多,所以对于积极评论的预测结果更准确,而其他情绪准确度依次下降。

然后，我们使用随机森林模型为每一条评论预测其情感倾向，我们还为数据集创建了一个新的字段，用来储存预测结果：

```
df1['Predicted_Sentiment'] = classifier.predict(vectorizer.transform(df1['Processed_Comment']))
```

用测结果分析：



从上面的图片中可以看到,模型对评论的预测结果会更加积极。

上图展示了各类评论的情感分布,可以看出对于剧情和特效这两个角度,消极评论极少,说明大多用户都对该电影的剧情和特效较为满意。

二、基于 LDA 主题模型的评论分类预测

模型的选择

数据集中没有每条评论主题，需要人工标注大量的评论来训练数据集，因此，为了方便起见，我们选择了无监督的 LAD 主题模型。

首先，我们从网上下载了停用词表，然后对评论进行分词，随后将评论文本转换成一个词频矩阵。

```
#主题提取
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import jieba
import re

def load_stop_words(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        stop_words = set(file.read().splitlines())
    return stop_words

chinese_stop_words = load_stop_words("百度停用词表.txt")

# 假设您的DataFrame为df1，评论列名为'Comment'
comments = df1['Comment']

# 文本预处理
def preprocess(text):
    # 移除标点符号
    text = re.sub(r'[^\w\s]', '', text)
    # 使用jieba进行中文分词
    words = jieba.cut(text)
    # 移除停用词
    words = [word for word in words if word not in chinese_stop_words]
    return ' '.join(words)

# 应用文本预处理
processed_comments = comments.apply(preprocess)

# 向量化处理
vectorizer = CountVectorizer(max_df=0.95, min_df=2)
X = vectorizer.fit_transform(processed_comments)
```

随后，我们训练 LDA 模型并使用

```
# 定义LDA模型
lda = LatentDirichletAllocation(n_components=5, random_state=0)  # 可以调整n_components来改变主题数量

# 训练模型
lda.fit(X)

# 显示主题关键词
def print_top_words(model, feature_names, n_top_words):
    for topic_idx, topic in enumerate(model.components_):
        message = "Topic #%d: " % topic_idx
        message += " ".join([feature_names[i] for i in topic.argsort()[:-n_top_words - 1:-1]])
        print(message)

print_top_words(lda, vectorizer.get_feature_names(), 10)
```

然后，我们得出了第一次分类的结果

```
#Topic #0: 与3D效果、电影院观影体验等相关的评论。
#Topic #1: 似乎集中在特定的超级英雄（如绿巨人、钢铁侠）和相关的情感或风格（如喜爱、搞笑）。
#Topic #2: 聚焦于超级英雄拯救地球或大片风格的主题。
#Topic #3: 强调电影的娱乐性，如好看、不错、爆米花电影、剧情和特效等。
#Topic #4: 围绕特定的超级英雄角色（如美国队长、钢铁侠、雷神）和他们在电影中的角色。
```

：

第一次主题建模

主题	词汇1	词汇2	词汇3	词汇4	词汇5	词汇6	词汇7	词汇8	词汇9	词汇10
体验	3d	效果	电影	电影院	imax	看过	不错	一个	片子	啊啊啊
剧情	绿巨人	钢铁	喜欢	洛基	队长	搞笑	美国	联盟	可爱	英雄
主题	英雄	超级	美国	拯救	电影	地球	一个	世界	大片	外星人
娱乐	电影	好看	不错	爆米花	剧情	特效	场面	一部	过瘾	精彩
角色	美国	队长	大片	钢铁	雷神	鹰眼	黑寡妇	绿巨人	浩克	loki

第一次主题建模的结果如图所示,由于分类并不细化有些分类很相近,于是我们更改模型参数并再次进行主题建模：

```python
##第二次主题提取
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import jieba
import re

def load_stop_words(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        stop_words = set(file.read().splitlines())
    return stop_words

chinese_stop_words = load_stop_words("百度停用词表.txt")

# 假设您的DataFrame为df1，评论列名为'Comment'
comments = df1['Comment']

# 文本预处理
def preprocess(text):
    # 移除标点符号
    text = re.sub(r'[^\w\s]', '', text)
    # 使用jieba进行中文分词
    words = jieba.cut(text)
    # 移除停用词
    words = [word for word in words if word not in chinese_stop_words]
    return ' '.join(words)

# 应用文本预处理
processed_comments = comments.apply(preprocess)

# 使用TF-IDF向量化
tfidf_vectorizer = TfidfVectorizer(max_df=0.85, min_df=2, ngram_range=(1, 2))
X_tfidf = tfidf_vectorizer.fit_transform(processed_comments)
```

```
# 使用调整后的参数重新训练LDA模型
lda_tfidf = LatentDirichletAllocation(n_components=10, learning_method='online', random_state=0)
lda_tfidf.fit(X_tfidf)
# 训练模型
lda.fit(X)

# 显示主题关键词
def print_top_words(model, feature_names, n_top_words):
    for topic_idx, topic in enumerate(model.components_):
        message = "Topic #%d: " % topic_idx
        message += " ".join([feature_names[i] for i in topic.argsort()[:-n_top_words - 1:-1]])
        print(message)

print_top_words(lda, vectorizer.get_feature_names(), 10)




# 重新打印主题关键词
print_top_words(lda_tfidf, tfidf_vectorizer.get_feature_names(), 10)
```

得出的分类结果如下所示：

第二次主题建模

主题	词汇1	词汇2	词汇3	词汇4	词汇5	词汇6	词汇7	词汇8	词汇9	词汇10	词汇11	词汇12
剧情	好看	英雄	超级	超级	英雄	拯救	世界	电影	幽默	商业片	地球	
特效	3d	可爱	效果	电影院	imax	刺激	看着	绿巨人	可爱	3d	效果	绿巨人
角色喜爱	漫威	电影	看过	五星	真的	好萌	啊啊啊	抖森	热血沸腾	亮点		
角色	钢铁	队长	美国	队长	美国	欢乐	绿巨人	loki	雷神	鹰眼	黑寡妇	
娱乐性质	爆米花	电影	爆米花	电影	搞笑	睡着	精彩	很爽	场面	一部	彩蛋	
期待	哈哈哈	hulk	期待	碉堡	妇联	想象	大乱斗	续集	蜘蛛侠	超人		
感受	不错	绿巨人	喜欢	大片	大杂烩	英雄主义	剧情	美国	一个	3d		
唐尼	过瘾	热闹	唐尼	罗伯特	罗伯特	唐尼	英雄	帅哥	票价	简单	很棒	
特效	钢铁	浩克	特效	喜欢	感觉	绿巨人	喜欢	钢铁	斯嘉丽	美队	无敌	
特征	洛基	联盟	high	第一部	hero	复仇者	一星	无尿点	复仇者	联盟	喜剧片	

接下来,我们使用 LAD 模型,对电影复仇者联盟的所有评论进行分类,

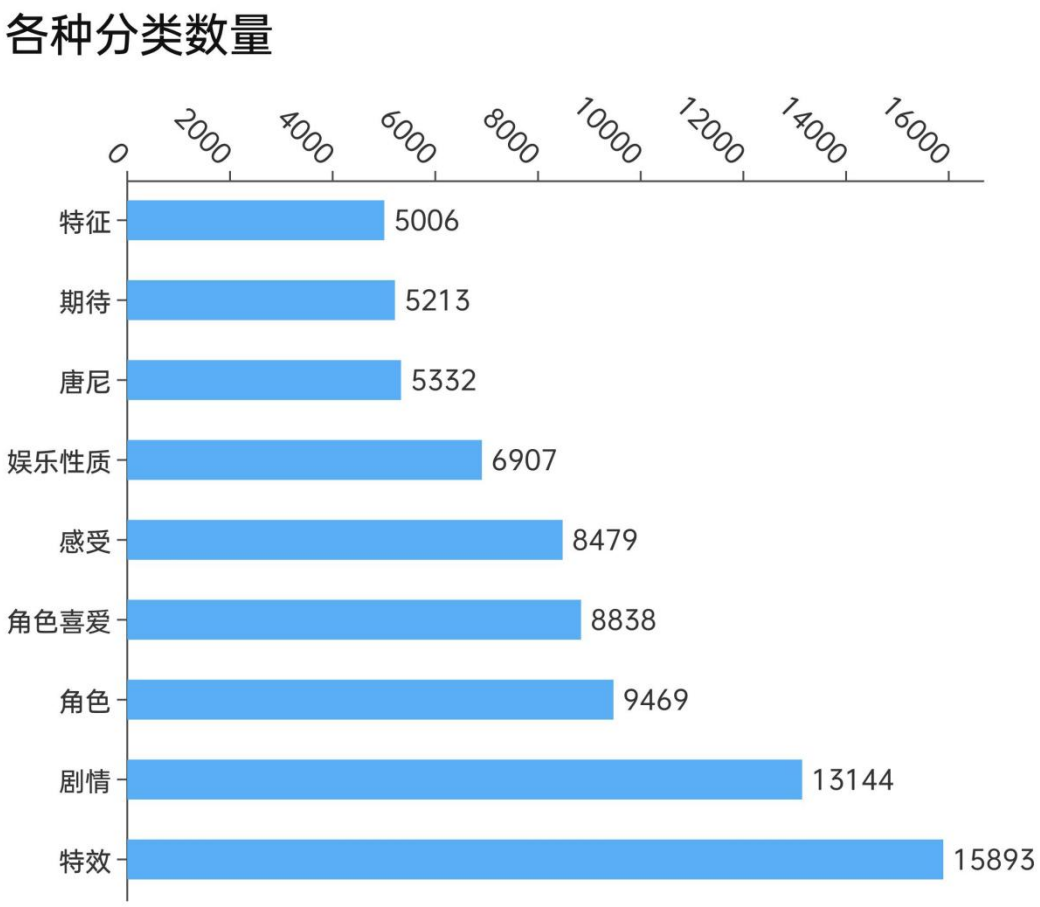
使用训练好的 LDA 模型对文本数据集进行变换。从而获取每个主题在该评论中出现的概率。

```
# 首先，我们获取每条评论的主题分布
topic_distribution = lda_tfidf.transform(X_tfidf)
```

通过查找每条评论的主题分布中概率最高的主题来确定其主要主题

```
# 接下来，为每条评论选择最可能的主题
# 这将是概率最高的主题
main_topic = topic_distribution.argmax(axis=1)

# 将主题标签添加到原始DataFrame中
df1['Topic_Label'] = main_topic

# 将数字标签转换为更具描述性的标签
topic_names = {0: '剧情', 1: '特效', 2: '角色喜爱', 3: '角色', 4: '娱乐性质',
               5: '期待', 6: '感受', 7: '唐尼', 8: '特效', 9: '特征'}
df1['Topic_Label'] = df1['Topic_Label'].map(topic_names)
```

分类完成后,我们对各种类别进行了统计分析,得出如下结果:



可以看出,特效,剧情是观众最关注的部分。观众对于这几部分的讨论最多。

三、基于自定义文本挖掘算法的受欢迎的角色、元素倾向统计

首先，我们通过创建角色映射表,即分别能够代表每个角色的关键词，来找到每个评论都提到了哪个角色。

由于电影《复仇者联盟》角色众多，且电影较为热门所以评论数据也多，于是我们以电影《复仇者联盟》为例进行了这一个文本挖掘方法。通过网上查找资料，我们得出了如图所示的角色映射表：

角色映射表

角色	关键词1	关键词2	关键词3	关键词4	关键词5	关键词6	关键词7	关键词8
钢铁侠	小罗伯特·唐尼	托尼·斯塔克	托尼	唐尼	铁人	钢铁侠		
美国队长	克里斯·埃文斯	史蒂芬·罗杰斯	史蒂夫·罗杰斯	美队	罗杰斯	埃文斯	美国队长	
雷神	克里斯·海姆斯沃斯	索尔·奥丁森	索尔	海姆斯沃斯	雷神			
绿巨人	马克·鲁弗洛	布鲁斯·班纳	班纳	浩克	绿巨人	鲁弗洛		
黑寡妇	斯嘉丽·约翰逊	娜塔莎·罗曼诺夫	娜塔莎	罗曼诺夫	斯嘉丽	黑寡妇	寡妇	汤包
鹰眼	杰瑞米·雷纳	克莱特·巴顿	巴顿	雷纳	鹰眼			
洛基	汤姆·希德勒斯顿	洛基·劳斐森	劳斐森	希德勒斯顿	洛基	抖森	丁小基	基妹
尼克·弗瑞	塞缪尔·杰克逊	尼古拉斯·约瑟夫·弗瑞	弗瑞	杰克逊				

具体步骤如下:

导入库:

我首先导入了 pandas 和 matplotlib.pyplot 库。pandas 用于数据处理和分析，而 matplotlib.pyplot 用于数据可视化。

读取数据:

我使用 pandas 的 read_excel 函数读取了一个名为'情感倾向预测结果.xlsx'的 Excel 文件，并将其存储在变量 df1 中。这个文件包含电影评论数据及其预测的情感倾向。

创建角色别名映射表:

我创建了一个名为 character_aliases 的字典，用于存储电影角色及其别名。例如，钢铁侠有多个别名，如“托尼”、“唐尼”等。

生成反向映射表:

我创建了一个名为 reverse_aliases 的字典，这是 character_aliases 的反转映射，用于将评论中的别名映射回标准角色名称。

标准化评论中的角色提及：

我定义了一个名为 standardize_character_mentions 的函数，该函数接收一条评论作为输入，并使用 reverse_aliases 映射表将其中的角色别名替换为标准名称。

应用标准化函数：

我将 standardize_character_mentions 函数应用于 df1 中的每条评论，以确保所有的角色提及都被标准化，并将结果存储在新列 Standardized_Comment 中。

统计每个角色的情感提及次数：

我创建了一个名为 character_sentiments 的字典，用于存储每个角色在不同情感类别（积极、中立、消极）下的提及次数。

我遍历每个角色和情感类别，计算在标准化的评论中提及该角色且具有相应情感倾向的评论数量。

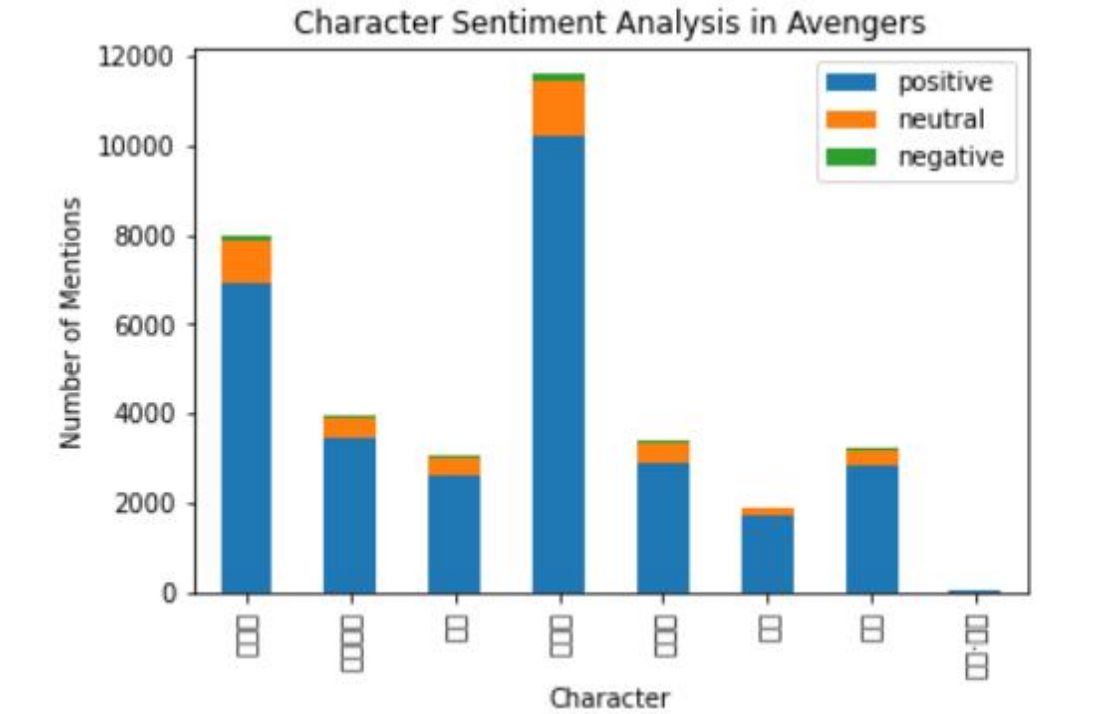
数据可视化：

我将 character_sentiments 字典转换为 pandas DataFrame，命名为 df_character_sentiments，以便于可视化。

最后，我使用 matplotlib 的 plot 函数绘制了一个条形图，展示了每个角色在不同情感类别下的提及次数。我设置了 x 轴标签为“Character”，y 轴标签为“Number of Mentions”，并给图表添加了标题“Character Sentiment Analysis in Avengers”。

```python
import pandas as pd
import matplotlib.pyplot as plt


df1 = pd.read_excel('情感倾向预测结果.xlsx')

character_aliases = {
    "钢铁侠": ["小罗伯特·唐尼", "托尼·斯塔克", "托尼", "唐尼", "铁人","钢铁侠"],
    "美国队长": ["克里斯·埃文斯", "史蒂芬·罗杰斯", "史蒂夫·罗杰斯", "美队", "罗杰斯", "埃文斯","美国队长"],
    "雷神": ["克里斯·海姆斯沃斯", "索尔·奥丁森", "索尔", "海姆斯沃斯","雷神"],
    "绿巨人": ["马克·鲁弗洛", "布鲁斯·班纳", "班纳", "浩克","绿巨人", "鲁弗洛"],
    "黑寡妇": ["斯嘉丽·约翰逊", "娜塔莎·罗曼诺夫", "娜塔莎", "罗曼诺夫", "斯嘉丽","黑寡妇","寡妇","汤包"],
    "鹰眼": ["杰瑞米·雷纳", "克莱特·巴顿", "巴顿", "雷纳","鹰眼"],
    "洛基": ["汤姆·希德勒斯顿", "洛基·劳斐森", "劳斐森", "希德勒斯顿","洛基","抖森","丁小基","基妹"],
    "尼克·弗瑞": ["塞缪尔·杰克逊", "尼古拉斯·约瑟夫·弗瑞", "弗瑞", "杰克逊"]
}

# 反转映射表，用于查找角色的标准名称
reverse_aliases = {alias: standard for standard, aliases in character_aliases.items() for alias in aliases}

# 标准化评论中的角色提及
def standardize_character_mentions(comment):
    for alias in reverse_aliases:
        if alias in comment:
            comment = comment.replace(alias, reverse_aliases[alias])
    return comment

# 应用标准化函数
df1['Standardized_Comment'] = df1['Comment'].apply(standardize_character_mentions)

# 统计每个角色的情感提及次数
character_sentiments = {character: {'positive': 0, 'neutral': 0, 'negative': 0} for character in character_aliases}
```

```python
for character in character_aliases:
    for sentiment in ['positive', 'neutral', 'negative']:
        # 对于每个角色和情感类型，计算提及次数
        character_sentiments[character][sentiment] = df1[
            (df1['Standardized_Comment'].str.contains(character)) &
            (df1['Predicted_Sentiment'] == sentiment)
        ].shape[0]

# 转换为DataFrame进行可视化
df_character_sentiments = pd.DataFrame(character_sentiments).T

# 绘制条形图
df_character_sentiments.plot(kind='bar', stacked=True)
plt.xlabel('Character')
plt.ylabel('Number of Mentions')
plt.title('Character Sentiment Analysis in Avengers')
plt.show()
```



对于最受欢迎的元素的分析,方法步骤和上面类似,我们采用的元素映射表为

最受欢迎的元素映射表

元素	关键词1	关键词2	关键词3	关键词4	关键词5	关键词6	关键词7	关键词8	关键词9
悬疑	谜团	不确定	迷雾	悬念	疑问	悬疑	神秘	难解	复杂
浪漫	爱情	情感	恋爱	情侣	甜蜜	浪漫	温馨	感人	情节
冒险	探险	旅行	勇敢	探索	奇遇	冒险	刺激	探险家	英勇
幽默	幽默	搞笑	笑料	欢乐	娱乐	喜剧	幸福	滑稽	有趣
动作	打斗	追逐	激烈	爆炸	速度	动作	紧张	冲突	战斗
惊悚	恐怖	紧张	恐惧	惊吓	悬念	惊悚	惊险	阴森	紧迫
科幻	未来	外星	科技	幻想	超自然	科幻	创新	探索	奇迹
剧情	故事	情节	角色	剧情发展	冲突	剧情	叙述	情感	发展
奇幻	魔法	神话	传说	幻境	奇异	奇幻	梦幻	非现实	神秘
悲剧	悲伤	失落	痛苦	哀伤	悲哀	悲剧	心痛	悲惨	苦难
演技	表演	演绎	角色塑造	表情	情感表达	演技	演出	扮演	塑造
视觉效果	特效	视觉	美术设计	摄影	画面	视觉效果	效果	视觉冲击	美学
音乐与音效	配乐	背景音乐	音效	声音设计	原声	音乐	旋律	节奏	声响
导演	导演	执导	视角	风格	导演风格	创作	监督	导演作品	指导
情感反应	感动	震撼	启发	共鸣	情感反应	感受	情绪	心灵触动	共感

5、结果分析与讨论

重点说明商务数据分析结果的实际意义和应用价值，说明数据分析结果对于实际的管理问题有哪些实际意义和决策支持作用。

（具体的分析结果在《商务数据统计分析报告》中有了完整的体现，该报告完整的对于每一个分析结果进行了一定的说明和实际应用，且存放了大量的可视化图表和动态图表。在此，我省略图表的展示，只进行文字描述）

基于观众喜好的个性化推荐

经过我们的分析，我们得出了各个评论对于某个角色或者某种电影元素的情感倾向，因此，可以利用这些分析结果，针对不同的用户，进行个性化推荐，例如，推荐他们喜爱的角色以及相应的演员的电影。或者利用这些洞见，电影发行商可以针对不同的用户个性化定制宣传材料，突出观众喜爱的角色或电影元素，以吸引更多观众。

垃圾评论的管理与过滤

我们分析了垃圾评论和总评论随着时间段变化趋势，得出垃圾评论和总评论的趋势基本相同，也就是说，在电影刚发布时，很容易出现垃圾评论，影评网站需要加强这个时间段的监管。平台可以在电影刚发布时期加强监管，采用自动化工具识别和过滤垃圾评论，从而维护健康的评论环境。

电影宣传策略优化

我们分析累计平均评分随时间的变化趋势，发现各个电影的累计平均评分随时间变化趋势不大，这可能是由于电影刚上映时的评论数量远超其他时刻，这也说明，电影商需要抓住刚上映的这段时间，做好营销和宣传，才能够获得较高的评分。因此，电影发行商可以在评论高峰期增加宣传力度，或在热度下降时推出特别活动，以维持电影的关注度和市场表现。

观众理性看待评分的建议

我们研究发现，刚上映时，电影的评价都会略高，这可能是因为电影发行方通过海报等各种方式宣传的影响，然后很快就会下降，因此观众需要电影刚上映的电影的评分持理性态度。

电影发布时间建议

我们研究发现，时间背景不同，观众对于电影的评价也会有所变化.这是符合常识的，因此，电影的发行也需要选择一个合适的时间段.这也说明了为什么抗美援朝题材电影经常选择在国庆档上映。

基于观众评价的电影改进意见

我们研究发现积极情感和中立情感的讨论的角度有很大不同，四星和五星评论的讨论角度有很大不同，这可能是导致他们没有给出 5 分或者没有表现出积极情感的原因，电影发行商可以从这些数据中寻找关于电影的修改意见。例如我们发现 4 星的评论相较于 5 星评论更乐意与讨论剧情和特效，因此，可能剧情和特效是到这这部分人没有给出 5 星的原因。

海报、周边设计、后续作品演员选择

通过分析，我们得出了上图所示的这些结果.大致得出最受欢迎的角色是绿巨人，其次是钢铁侠，因此电影发行商可以设计相关的海报和周边等。也可以选择这些受欢迎的演员来继续合作。

垃圾评论的影响

我们发现，在评分中，垃圾评论的平均评分和所有评论大致相同，但是垃圾评论的平均评分普遍略高于平均评分.由此得出，垃圾评论倾向于给较高的评分，这会使得电影评分增高。识别垃圾评论的趋势，特别是它们倾向于给出更高的评分，有助于电影评价平台维护评分的真实性和可信度。电影评价平台可以加强对垃圾评论的过滤和监管，确保电影的评分反映真实的观众反馈。

评论数量的变化

我们分别针对单个电影进行了分析，左上角图片展示了个各电影的评论数量随时间变化趋势，可以看到每部电影都是刚上映的时候，受到最多的评论，随后热度很快下降。且刚发布时的评论数量远高于其他时间的评论数量。评论数量在电影上映初期的激增和随后的快速下降反映了观众对新上映电影的高度关注和随后的关注度下降。电影制片方和发行商应在电影上映初期集中宣传力度，利用这一时期的高关注度提升电影知名度。

评论的统计性分析

我们研究发现，2 星的评论最容易受到点赞，而四星，五星评论的数量最多.在不同电影中发布评论，受到点赞的难易程度差别显著。不同星级评论受欢迎程度的差异揭示了观众对评论内容的不同反应。电影评论平台可据此优化评论展示策略，如突出展示更有参考价值的评论。

电影热度分析

我们分析发现，均分越高的电影一般总评论数量越多，即热度越高，电影发行商可根据这些数据调整市场策略，提升电影在市场上的竞争力。

电影行业趋势分析

我们发现，疯狂动物城，大圣归来等电影具有较强的持续影响力，热度经久不衰，也可以看出在所有电影中，发布越晚的电影，总热度一般越高，可能是因为人们生活水平的提高使得人们的精神追求更加的丰富了，也说明电影行业是一个朝阳行业，具有向上的生长力。电影产业应投资于新内容的创作和宣传，同时关注市场的长期趋势和变化。

电影持续影响力分析

以电影《复仇者联盟》未分析对象，我们发现，随着时间变化，在某一时期内，中立和消极情感、较低星级的评论会有所增加，而角色喜爱这个讨论角度变化趋势极大，经常跃居第三又退至倒数，而关于特效和剧情的评论数量你追我赶，经常轮流当做第一和第二。电影制片方可根据这些趋势优化电影内容和宣传策略，以维持或提升其长期影响力。