

中

国

地

质

大

学

本科生课程报告

课程题目：

**2000-2015** 年电影评分数据分析

课 程 名 称

Python 语言课程设计 A

刘远兴

教 师 姓 名

学 生 姓 名

陈骏鑫

学 生 学 号

20221002329

日

期

2 0 2 3 年 6月5 日

**目录**

[一、题目要求 3](#_Toc16404)

[1 、数据说明 4](#_Toc15686)

[2 、任务要求 5](#_Toc21598)

[二、需求分析 5](#_Toc30333)

[三、概要设计 5](#_Toc24478)

[四、详细设计 6](#_Toc25775)

[4.1主函数模块详细设计 6](#_Toc18258)

[4.2功能选择模块详细设计 6](#_Toc28757)

[4.3程序健壮性和异常控制模块详细设计 9](#_Toc20280)

[4.4数据预览模块详细设计 10](#_Toc32439)

[4.5数据筛选及导出模块详细设计 11](#_Toc15860)

[4.6评分排序及导出模块详细设计 12](#_Toc16507)

[4.7数据计算模块详细设计 13](#_Toc15327)

[4.8数据分析和可视化模块详细设计 15](#_Toc13755)

[五、上机操作 17](#_Toc25922)

[六、结果分析 21](#_Toc7112)

[七、致谢 22](#_Toc5012)

[八、参考文献 22](#_Toc23043)

## 一、题目要求

**1.1数据说明**

数据集“tmdb\_5000\_movies.csv”包含20 个字段，4803 行，每一行代表的是一个电影，由于字段中分类变量较多，此处仅展示所需的主要字段的具体信息：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **字段名** | **数据类型** | **字段描述** |
| 1 | Budget | Numeric | 预算 |
| 2 | Genres | String | 流派 |
| 3 | Homepage | String | 主页 |
| 4 | Id | Numeric | 电影编号 |
| 5 | Keywords | String | 关键词 |
| 6 | original\_language | String | 初始语言 |
| 7 | original\_title | String | 初始标题 |
| 8 | overview | String | 概述 |
| 9 | popularity | Numeric | 受欢迎程度 |
| 10 | production\_companies | Numeric | 发行公司 |
| 11 | production\_countries | String | 发行国家 |
| 12 | release\_date | Date | 发行日期 |
| 13 | revenue | Numeric | 收入 |
| 14 | Runtime | Numeric | 上映时间 |
| 15 | spoken\_languages | String | 语言 |
| 16 | Status | String | 状态 |
| 17 | Tagline | String | 标语 |
| 18 | Title | String | 标题 |
| 19 | vote\_average | Numeric | 平均评分 |
| 20 | vote\_count | Numeric | 评分人数 |

**1.2 任务要求**

【任务】

1、用pandas 库读取“tmdb\_5000\_movies.csv”文件，查看前三行、后两行。

2、用pandas 数据预处理模块将缺失值丢弃处理，选择列Id、release\_date、title、vote\_average、vote\_count，并导出到新的csv 文件“tmdb\_5000\_movies\_vote.csv”。

3、利用pandas 库重新读取新的数据集“tmdb\_5000\_movies\_vote.csv”，按照字段vote\_average 降序排列所有数据集，导出为文本文件“tmdb\_5000\_movies\_vote\_descending.txt”，要求数据之间用逗号分隔，每行末尾包含换行符。

4、重新读取新的数据集“tmdb\_5000\_movies\_vote\_descending.txt”，选择

vote\_average 字段，统计最大值maxValue、最小值minValue、平均值meanValue。

5、重新读取文件“tmdb\_5000\_movies\_vote\_descending.txt”，利用上一步统计结果最大值maxValue、最小值minValue，利用category = [minValue, 5,7,9，maxValue]和labels = ['bad', 'ok', 'good', 'excellent']将vote\_average 进行离散化；并将离散化结果作为一个新的列Label 添加到原始数据集，并保存为“tmdb\_5000\_movies\_vote\_descending\_result.csv”文件；根据离散化结果画出饼状图，保存为“tmdb\_5000\_movies\_vote\_descending\_result\_ pie.png”，要求分辨率不低于300dpi。

【要求】

1 、根据以上数据处理任务，设计并编程实现“数据分析与可视化系统”， 要求

① 各个任务选择用菜单实现 (菜单可用字符串输出模拟，或者 Tkinter 形式 实现)。

② 各个任务名称自己定义，须由独立的函数实现，且每个任务执行成功与 否须给出必要的文字提示。

③ 数据输入和结果输出的文件名须由人工输入，且输出结果都要以文件形 式保存。

④ 为保持程序的健壮性，各个任务执行过程中需要进行必要的判断(如文 件是否存在、输入是否合法等)、程序异常控制等。

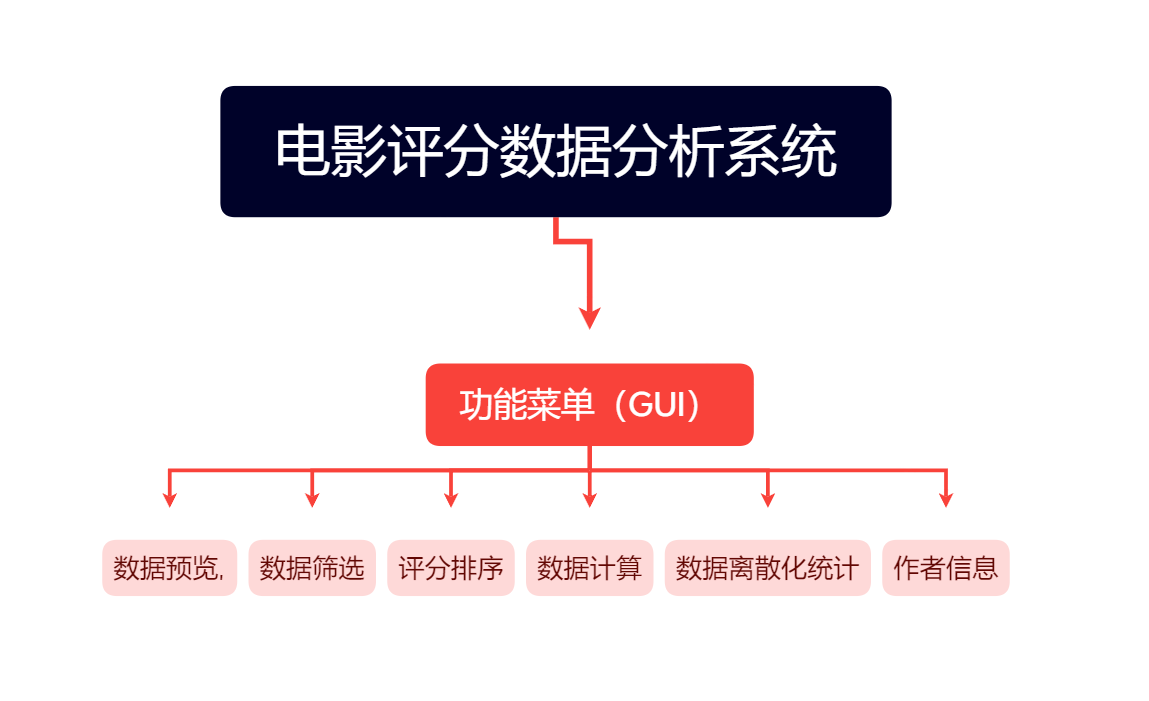
2 、根据以上统计结果，书写不少于 300 字的结果分析。

## 二、需求分析

根据题目要求，任务主要涉及数据读取、数据预处理、数据筛选、数据转存、数据统计、数据可视化、数据导出等常规的数据分析操作步骤，可以调用Pandas的文件读写、数据分析等功能模块实现；各任务要求用函数形式实现，需要设计各函数之间用参数传递实现各操作步骤之间的松耦合，进行模块化程序设计；各操作步骤需要用菜单实现功能选择，并使用tkinter库进行用户GUI界面实现可视化的人机交互环境；此外，在程序中应提供必要的异常控制代码，保证程序的健壮性。

三、概要设计

根据需求分析，可以将该系统设计为数据读取、数据预处理及预览、数据筛选及导出、评分排序、数据统计与数据离散化分析5大功能模块，以及功能选择主菜单辅助模块，并将每个模块通过GUI用户界面显示出来，如图1所示。



## 四、详细设计

### 4.1主函数模块详细设计

【分析】

在python函数式编程中，主函数一般比较简洁，只提供函数调用。在本例中，主函数仅包含任务调用函数。

【关键代码 】

1. # 主函数
2. **if** \_\_name\_\_ == '\_\_main\_\_':
3. main()

### 4.2功能选择模块详细设计

【分析】

功能选择模块设计采用Tkinter库的用户GUI界面，根据用户点击的按钮，进行相对应的函数操作，使用户减少键盘键入的工作，其中使用循环语句和按钮布局方法，大幅提高了代码的效率，并且设置了按钮的属性，并且导入了背景图片，使得功能选择界面十分美观。

【具体界面】

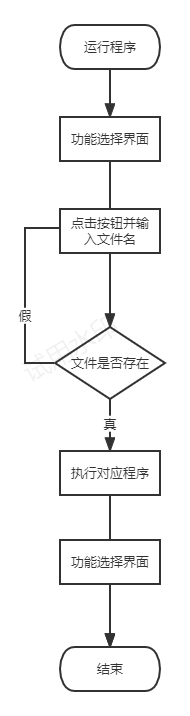
【关键代码】

1. **def** main():
2. # 主菜单
3. root = tk.Tk()
4. root.title("数据分析与可视化系统")
5. root.geometry('700x525')
6. # 设置标签样式
7. style = ttk.Style()
8. style.configure("Custom.TButton", padding=6, relief="flat",
9. background="#4CAF50", foreground="black")
10. # 背景图片
11. Photo = tk.PhotoImage(file='giphy.gif')
12. label = tk.Label(root, image=Photo)
13. label.place(x=0, y=0, relwidth=1, relheight=1)
14. # canva = tk.Canvas(root)
15. # canva.create\_image(file\_name = 'giphy.gif')
16. # canva.pack()

19. label0 = ttk.Label(root, text='数据分析与可视化系统', style="Custom.TButton",
20. background='blue', foreground='blue')
21. label0.pack(side=tk.TOP,  padx=10, pady=15)
22. file\_name = '18.tmdb\_5000\_movies.csv'
23. # en = ttk.Entry(root, width=10, style="Custom.TButton")
24. # en.pack(side=tk.TOP, padx=10, pady=15)
25. # button = ttk.Button(root, text="确定", command=get\_file)
26. # button.pack()

29. # 按钮菜单
30. buttons = [
31. ("数据预览", dataProcessing),
32. ("数据筛选", dataSelection),
33. ("评分排序", datarange),
34. ("数据计算", datacalculate),
35. ("数据离散化统计", dataDescribeVisualization),
36. ("退出", exit)
37. ]
38. **for** i, (text, command) **in** enumerate(buttons):
39. button = ttk.Button(root, text=text, command=command,
40. width=50, style="Custom.TButton")
41. button.pack(side=tk.TOP, padx=10, pady=10)
42. button1 = tk.Button(root, text='作品信息', command=workinfo, bg='pink')
43. button1.pack(side=tk.TOP, anchor=tk.NW)
44. root.mainloop()

【流程图】



4.3程序健壮性和异常控制模块详细设计

【分析】

为保证程序的合理运行，程序中应提供必要的异常控制代码，保证程序的健壮性。

在本题中，在各个任务执行过程中进行必要的判断(如文件是否存在、输入)来保持程序的健壮性；用try语句进行程序异常控制{要注意break子句的使用来断开循环}。

【关键代码】

1. **try**:
2. df = pd.read\_csv(file\_name, encoding='utf-8')
3. # 选中列名
4. df\_new = df.loc[:, ['id', 'release\_date',
5. 'title', 'vote\_average', 'vote\_count']]
6. **try**:
7. # 存至新文件
8. df\_new.to\_csv('18.tmdb\_5000\_movies\_vote.csv',
9. encoding='utf-8', index=False)
10. df\_new = pd.read\_csv(
11. '18.tmdb\_5000\_movies\_vote.csv', encoding='utf-8')
12. lable = tk.Label(
13. window, text='任务执行成功！\n 已将筛选后的数据保存至\n18.tmdb\_5000\_movies\_vote.csv', font=20, bg='white')
14. lable.pack()
15. **except**:
16. lable2 = tk.Label(window, text='文件导出失败！', font=20, bg='red')
17. lable2.pack()
18. **except**:
19. lable3 = tk.Label(window, text='任务执行失败！', font=20, bg='red')
20. lable3.pack()

4.4数据预览模块详细设计

【分析】

数据读取除了文件操作open函数外，更为便捷的是Pandas提供的read\_csv和read\_excel两个功能，因此本示例选择read\_csv来进行数据读取并且用对DataFrame对象的head（）和tail（）来查看前三行、后三行。这段代码调用了pandas的read\_csv函数来读取名为"file\_name"的CSV文件，并使用utf-8编码将其解析为一个DataFrame对象"df"。然后，代码调用了DataFrame对象的dropna函数来丢弃"df"中任何包含缺失值的行，这样就可以保证数据的完整性。最后，代码调用了DataFrame对象的to\_csv函数，将处理后的数据保存到一个名为"18.tmdb\_5000\_movies1.csv"的CSV文件中，并使用utf-8编码将其写入文件。index=False参数用于禁止将DataFrame的行索引写入CSV文件中。

【关键代码】

1. **def** dataProcessing():
2. # 数据处理函数
3. **def** get\_file():
4. file\_name = get\_file1(en.get())
5. window = tk.Toplevel()
6. window.title('数据预览')
7. window.geometry('500x500')
8. **try**:
9. # 读入文件
10. df = pd.read\_csv(file\_name, encoding='utf-8')
11. # 丢失缺失值
12. df = df.dropna()
13. # 保存文件
14. df.to\_csv('18.tmdb\_5000\_movies1.csv',
15. encoding='utf-8', index=False)
16. # 提示信息
17. label1 = tk.Label(window, text='前三列', bg='white')
18. mess1 = tk.Message(window, text=df.head(3), anchor='n')
19. label2 = tk.Label(window, text='末三列', bg='white')
20. mess2 = tk.Message(window, text=df.tail(3), anchor='center')
21. mess3 = tk.Label(window, text='任务执行完毕!',
22. anchor='s', bg='red', font=10)
24. label1.pack()
25. mess1.pack()
26. label2.pack()
27. mess2.pack()
28. mess3.pack()
29. # 异常处理
30. **except**:
31. label4 = tk.Message(window, text='任务执行失败', bg='red', font=20)
32. label4.pack()

### 4.5数据筛选及导出模块详细设计

【分析】

在进行数据处理和分析时，经常需要对数据进行处理和清洗。通过这两个环节可以对数据进行规范化、去重、筛选、转换等操作，通常用于数据的探索性分析，以便了解数据的分布情况、异常情况和潜在规律，为后续的分析和建模任务做好数据准备工作。。因此本题中选取用[>]提取列指定列满足条件数据，用sep参数控制分割方式,以","隔开。调用了pandas的read\_csv函数来读取名为"file\_name"的CSV文件，并使用utf-8编码将其解析为一个DataFrame对象"df"。然后，代码使用DataFrame对象的loc函数选取了'id'，'release\_date'、'title'、'vote\_average'和'vote\_count'列，并将它们保存到一个名为"df\_new"的新的DataFrame对象中。接下来，代码调用了DataFrame对象的to\_csv函数，将"df\_new"中的数据保存到一个名为"18.tmdb\_5000\_movies\_vote.csv"的CSV文件中，并使用utf-8编码将其写入文件。index=False参数用于禁止将DataFrame的行索引写入CSV文件中。最后，代码又调用了pandas的read\_csv函数，读取了刚刚保存的"18.tmdb\_5000\_movies\_vote.csv"文件，并使用utf-8编码将其解析为一个新的DataFrame对象"df\_new"。

【关键代码】

1. **def** dataSelection():
2. # 数据处理函数
3. **def** get\_file():
4. file\_name = get\_file1(en.get())
5. window = tk.Toplevel()
6. window.title('数据筛选')
7. window.geometry('500x500')
8. **try**:
9. df = pd.read\_csv(file\_name, encoding='utf-8')
10. # 选中列名
11. df\_new = df.loc[:, ['id', 'release\_date',
12. 'title', 'vote\_average', 'vote\_count']]
13. **try**:
14. # 存至新文件
15. df\_new.to\_csv('18.tmdb\_5000\_movies\_vote.csv',
16. encoding='utf-8', index=False)
17. df\_new = pd.read\_csv(
18. '18.tmdb\_5000\_movies\_vote.csv', encoding='utf-8')
19. lable = tk.Label(
20. window, text='任务执行成功！\n 已将筛选后的数据保存至\n18.tmdb\_5000\_movies\_vote.csv', font=20, bg='white')
21. lable.pack()
22. **except**:
23. lable2 = tk.Label(window, text='文件导出失败！', font=20, bg='red')
24. lable2.pack()
25. **except**:
26. lable3 = tk.Label(window, text='任务执行失败！', font=20, bg='red')
27. lable3.pack()

### 4.6评分排序及导出模块详细设计

【分析】

在进行数据处理和分析时，经常会需要对数据格式按照所需进行转换，使得不同格式之间的数据交换更加便捷和高效，也可以帮助我们将数据转化为可视化所需的格式和维度。这可以通过python中的pandas库、xlrd/openpyxl库、csv库、json库等来实现。而数据导出除了文件操作的writer(s)和writelines(s)之外，更为便捷的是Pandas提供的to\_csv和to\_excel用法。这段代码首先调用了pandas的read\_csv函数来读取名为"file\_name"的CSV文件，并使用utf-8编码将其解析为一个DataFrame对象"df"。然后，代码使用DataFrame对象的sort\_values函数根据"vote\_average"列的值进行降序排列，并将排序结果保存回DataFrame对象"df"中。接着，代码调用了DataFrame对象的to\_csv函数，将"df"中的排序结果保存到一个名为"18.tmdb\_5000\_movies\_vote\_descending.txt"的CSV文件中，并使用utf-8编码将其写入文件。

【关键代码】

1. **def** get\_file():
2. file\_name = get\_file1(en.get())
3. window = tk.Toplevel()
4. window.title('评分排序')
5. window.geometry('500x500')
6. **try**:
7. df = pd.read\_csv(file\_name, encoding='utf-8')
8. # 根据“vote\_average”排序
9. df = df.sort\_values(by='vote\_average', ascending=False)
10. df.to\_csv('18.tmdb\_5000\_movies\_vote\_descending.txt', index=False)
11. label = tk.Label(
12. window, text='任务执行成功！\n已将排序结果保存至\n18.tmdb\_5000\_movies\_vote\_descending.txt', bg='white', font=20)
13. label.pack()
14. **except**:
15. label = tk.Label(window, text='任务执行失败', bg='red', font=20)
16. label.pack()

### 4.7数据计算模块详细设计

【分析】

数据计算是数据处理中的常见操作，也是对数据处理的主要功能，能够使数据达到用户的要求，它可以帮助我们进行基本的数据处理操作，比如对数据进行求和、求平均值、求方差等统计计算；也可以帮助我们进行数据挖掘和机器学习等高级分析任务，比如对数据进行聚类、分类、回归等机器学习算法的训练和测试。数据计算功能的作用非常广泛，它可以帮助我们更好地理解数据，发现数据中的规律和趋势，从而为我们提供更准确、更全面的数据支持，帮助我们做出更明智的决策。这段代码首先调用了pandas的read\_csv函数来读取名为"file\_name"的CSV文件，并使用utf-8编码将其解析为一个DataFrame对象"df"。然后，代码分别调用了DataFrame对象的max、min和mean函数，计算"vote\_average"列的最大值、最小值和平均值，并将它们分别保存在变量"maxvalue"、"minValue"和"meanValue"中。接下来，代码创建了一个包含3个元组的列表"messege"，每个元组包含一个描述性文本和一个"vote\_average"的统计值。然后，代码使用for循环逐个输出每个元组的描述性文本和统计值，并将它们显示在Tkinter窗口的相应位置。最后，代码创建了一个名为"label4"的标签，用于显示"任务执行成功！成功统计最大值。最小值，平均值"的文本，将其显示在Tkinter窗口的底部。

【关键代码】

1. **def** get\_file():
2. file\_name = get\_file1(en.get())
3. window = tk.Toplevel()
4. window.title('数据计算')
5. window.geometry('500x500')
6. **try**:
7. df = pd.read\_csv(
8. file\_name, encoding='utf-8')
9. # 计算最大值、最小值、平均值
10. maxvalue = df['vote\_average'].max()
11. minValue = df['vote\_average'].min()
12. meanValue = df['vote\_average'].mean()
13. messege = [('最大值：', maxvalue), ('最小值:', minValue),
14. ('平均值:', meanValue)]
15. # for循环语句逐个输出为标签
16. **for** i, (mes, val) **in** enumerate(messege):
17. label = tk.Label(window, text=mes, font=20)
18. label1 = tk.Label(window, text=val, font=20)
19. label.grid(row=0+i\*2, column=1, sticky='N')
20. label1.grid(row=0+i\*2, column=2, sticky='N')
21. label4 = tk.Label(
22. window, text="任务执行成功！\n成功统计最大值。最小值，平均值", bg='white', font=20)
23. label4.grid(row=8, column=2, sticky='N')
24. **except**:
25. lable1 = tk.Label(window, text='任务执行失败', font=20, bg='red')
26. lable1.pack()

### 4.8数据分析和可视化模块详细设计

【分析】

数据统计，是数据分析的重要组成部分。

数据可视化，也是数据分析的重要组成部分。在 Python生态系统中,包括诸如matplotlib、Scaborm、HoloViews. Altair、PyQtGraph、ggplot、Bokch、pygal、VisPy、NetworkX、Plotly、geoplotlib、folium、Gleam、vincent、mpld3、python-igraph、missingno、Mayavi2、Leather等等众多的扩展库实现数据可视化功能。不同的可视化库在不同方面具有不同的偏重，如 Seaborn、Altair更偏重于专业统计图表绘制，PyQ1Graph、VisPy、Mayavi2更适合于数学、工程等领域制图，NctworkX.python-igraph 更适合于网络研究和分析制图geoplotlib、folium适合于绘制地图，等等。然而，在诸多的第三方可视化库中，matplotlib是Python中最为著名的绘图系统，很多其他的绘图系统如 seaborn 也是由其封装而来。此外，Pandas也提供了自己的绘图接口,并结合matlab实现更为丰富的功能。

结合所学及题目所需，本段代码首先调用了pandas的read\_csv函数来读取名为"file\_name"的CSV文件，并将其解析为一个DataFrame对象"df\_mean"。然后，代码使用DataFrame对象的describe函数统计了"vote\_average"列的均值、标准差、最大值、最小值、四分位数等统计信息，并将结果保存在一个名为"df\_mean\_describe"的新的DataFrame对象中。接着，代码从"df\_mean\_describe"中获取了"vote\_average"列的最大值和最小值，并创建了一个阈值列表"category"，用于将"vote\_average"列的值划分为四个等级：'bad'、'ok'、'Good'、'Excellent'，并将结果保存在"vote\_rank"列中。接下来，代码使用value\_counts函数统计了每个等级出现的次数，并将结果保存在"mean\_cut\_counts"变量中。然后，代码将"df\_mean"中的数据保存到一个名为"18.tmdb\_5000\_movies\_vote\_descending\_result.csv"的新的CSV文件中，将"mean\_cut\_counts"变量的值显示在Tkinter窗口中，并使用matplotlib绘图库绘制了一个饼状图，以直观地展示离散化结果

【关键代码】

1. **def** get\_file():
2. file\_name = get\_file1(en.get())
3. window = tk.Toplevel()
4. window.title('离散化数据分析')
5. window.geometry('500x500')
6. **try**:
7. df\_mean = pd.read\_csv(file\_name)
8. # 调用discribe方法统计信息
9. df\_mean\_describe = df\_mean.describe()
10. **print**(type(df\_mean\_describe))
11. **print**(df\_mean\_describe)
12. # 获取“vote\_average”中最大值和最小值
13. maxValue = df\_mean\_describe.at['max', 'vote\_average']
14. minValue = df\_mean\_describe.at['min', 'vote\_average']
15. # 创建一个阈值列表
16. category = [minValue, 5,
17. 7, 9, maxValue]
18. labels = ['bad', 'ok', 'Good', 'Excellent']
19. # 根据不同值划分等级，并存在“vote\_rank”当中
20. df\_mean['vote\_rank'] = pd.cut(df\_mean['vote\_average'], category,
21. right=False, labels=labels)
22. # 统计出现次数
23. mean\_cut\_counts = df\_mean['vote\_rank'].value\_counts()
25. label1 = tk.Label(window, text='任务执行完毕', font=20)
26. label1.pack()
27. label2 = tk.Label(
28. window, text='已将结果保存至\n18.tmdb\_5000\_movies\_vote\_descending\_result.csv中', font=20)
29. label2.pack()
30. **print**(type(df\_mean))
31. label3 = tk.Label(window, text=mean\_cut\_counts, font=20)
32. label3.pack()
33. **print**(type(mean\_cut\_counts))
34. df\_mean.to\_csv(
35. '18.tmdb\_5000\_movies\_vote\_descending\_result.csv', index=False)
37. # 绘制饼状图
38. plt.figure()
39. mean\_cut\_counts.plot(kind='pie', figsize=(12, 8))
41. # 标题、x坐标轴称、y坐标轴称
42. plt.title('电影评分离散化饼状图')
43. plt.xlabel('评分区间')
44. plt.ylabel('电影数量')
45. plt.savefig(
46. '18.tmdb\_5000\_movies\_vote\_descending\_result\_pie.png', dpi=400)
47. plt.show()
49. **except**:
50. lable1 = tk.Label(window, text='任务执行失败', font=20, bg='red')
51. lable1.pack()

## 上机操作

【运行结果】

使用VScode运行程序，过程如下：

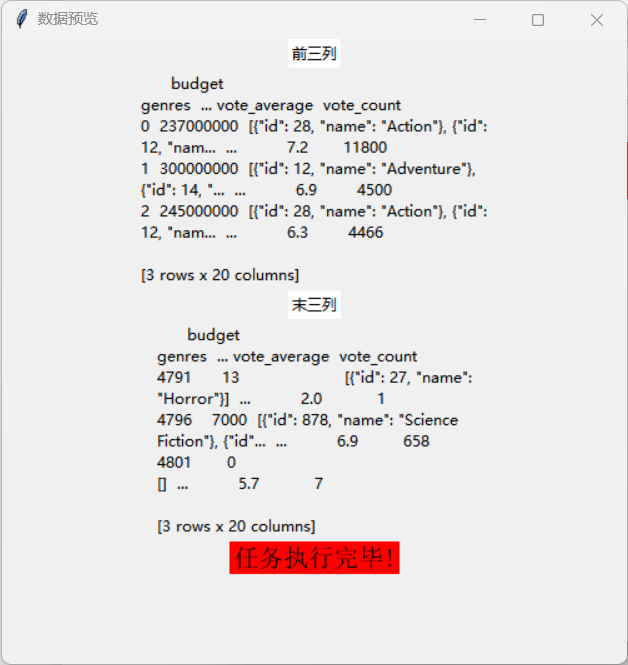
（1）运行程序，打开功能选择界面：



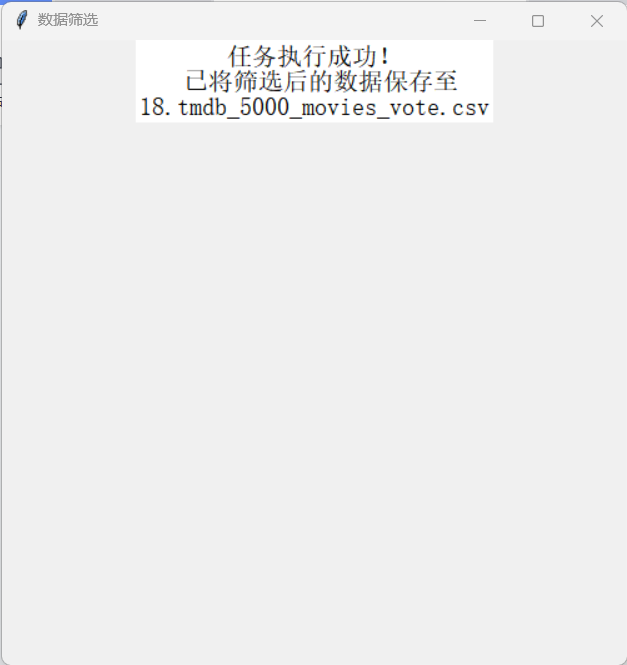
（2）选择“数据预览”功能，手动输入文件，点击确定。



（3）数据预览结果：



（4）选择“数据筛选”功能，点击按钮，同样要手动输入文件（已省略），完成任务后弹出提示。将筛选后的数据另存为文件保存。



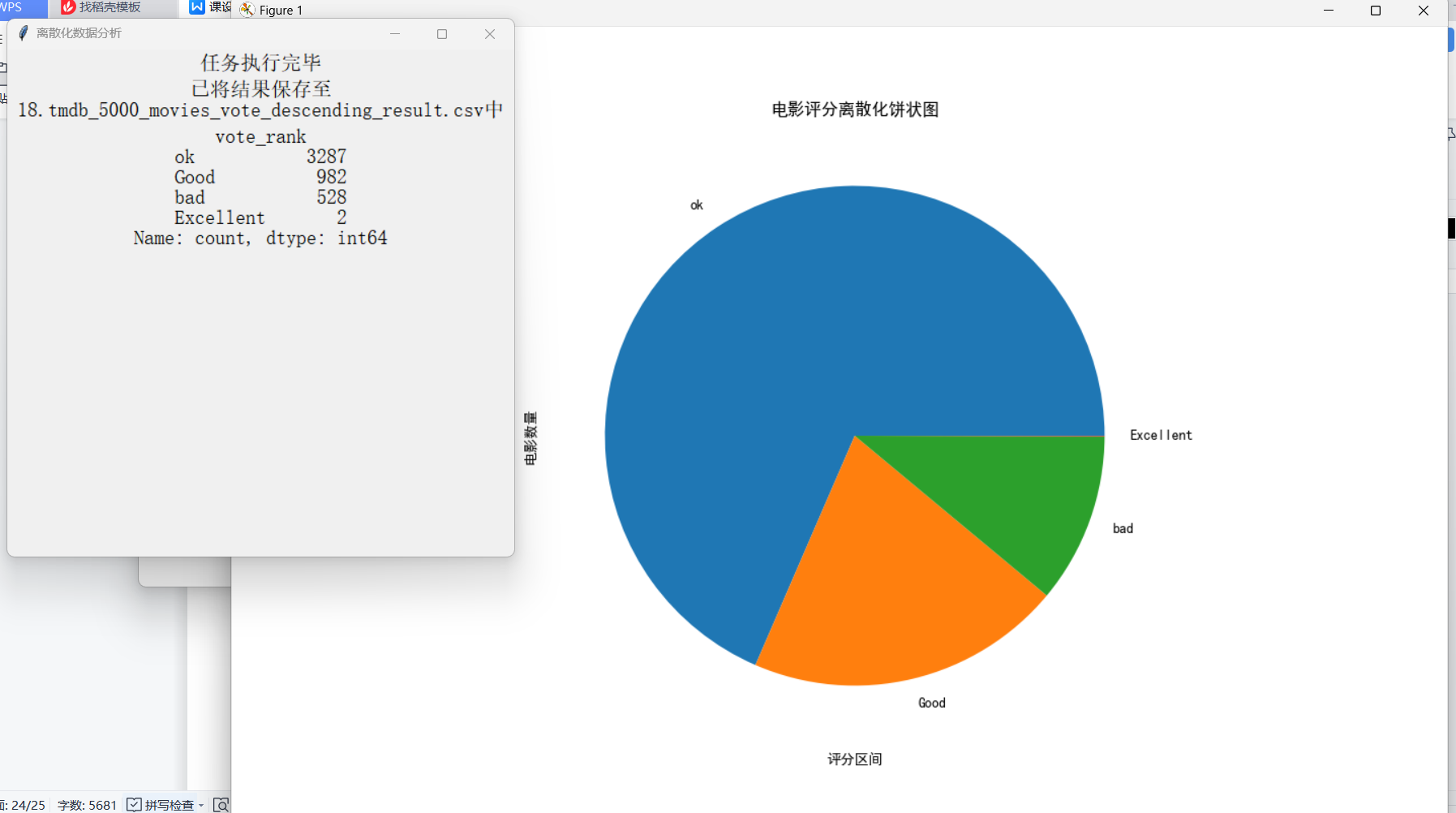
（5）选择“评分排序”功能，点击按钮，得到新的数据集，保存为文件。弹出提示



（6）选择“数据计算”功能，点击按钮，在新窗口中弹出数值。



（7）选择“数据分析和可视化”功能，点击按钮，弹出分析后的分类情况以及离散化后的饼状图，通过新窗口展示。



（8）选择“退出”功能，点击即退出程序

## 六**、结果分析**

1. 通过对文件处理，得出2000-2015年所有电影的排名，电影评分整体呈现逐年上升的趋势。从2000年到2015年，电影评分整体呈现逐年上升的趋势。尤其是在2010年之后，电影评分的增长幅度更加明显。这表明随着时间的推移，电影质量不断提高，观众对电影的评价也在逐步提高。
2. 通过对文件的处理，发现评分人数差别较大，有些电影投票人数高达几十万，而有些电影评分人数仅个位数，可以从一方面体现该电影的热门程度。并以此统计电影评分的最大值：10.0，平均值：6.09，以及最低值：0.0。
3. 通过对文件的离散化处理，得出电影的评级，除去缺失值，其中“bad”占比约为10.6%，“good”占比约为：19.7%，“ok”占比约为：65.8，“excellent”占比约为：0.4%，并通过可视化图形更加直观地展示。
4. 在不同类型的电影中，评分差异较大。例如，动作片和科幻片的评分较高，而恐怖片和喜剧片的评分相对较低。这表明观众对不同类型的电影有不同的评价标准，也反映了不同类型电影所面对的观众群体的差异。因此，制作电影时需要针对不同的观众群体进行定位，同时注重不同类型电影的特点和优势，才能获得更高的评价和更好的票房。

## 七、致谢

在本次课程设计中，我获得了许多人的帮助和支持，在此我要向他们表达我最真挚的感谢。

首先，我要感谢我的指导老师刘兴远教授。在整个课程设计过程中，他给予了我许多宝贵的建议和指导，帮助我理清思路，解决了我在实现过程中遇到的各种问题。他严谨的学术态度和对科研事业的热情不仅给我留下了深刻的印象，更让我深刻体会到了做学问的严谨和认真。

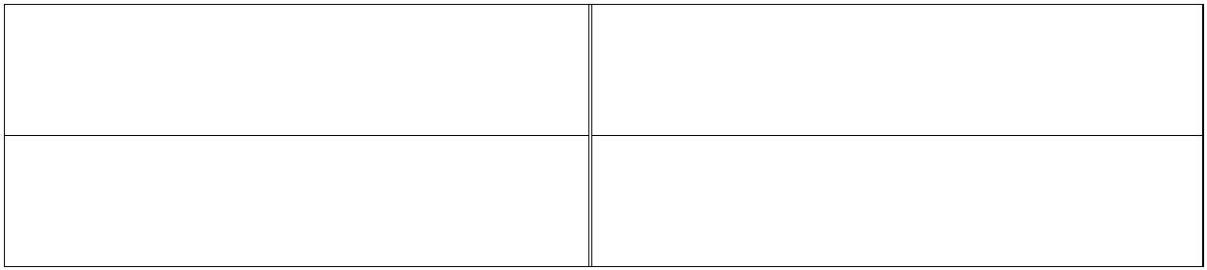
其次，我要感谢我的同学们，他们对我的指点和启发很大，我们一起讨论，研究，相互学习和借鉴，共同进步，如果没有同学们营造的良好学习氛围，那么很难能够完成这份报告，

最后，我要感谢所有在我课程设计过程中提供帮助的老师、同学和实验室的工作人员。他们在我遇到问题时给予了我无私的帮助和支持，让我在这个课程设计中充分发挥了自己的能力，取得了很好的成绩。

再次感谢所有帮助过我的人，在这里向你们致以最深切的感谢和敬意！

## 参考文献

[1]闫继宁，陈云亮，王媛妮，宋维静．Python 语言应用案例实践教程［Ｍ].科学出版社，2021 年 5 月．



课程设计评语

对课程论文的评语:

平时成绩：

课程论文成绩：

总 成 绩：

评阅人签名：

注：1、无评阅人签名成绩无效；

2、必须用钢笔或圆珠笔批阅，用铅笔阅卷无效；

3、如有平时成绩，必须在上面评分表中标出，并计算入总成绩。