

# 基于 Openpyxl 的高校期末成绩数据处理与分析方法研究

朱阁 胡硕娟

张家界航空工业职业技术学院 湖南 张家界 427000

**摘要:** 随着教育信息化的不断推进, 如何高效、准确地处理高校期末成绩数据成为一个重要问题。本文研究了基于 Python 和 Openpyxl 库的期末成绩数据处理方法, 设计并实现了一个自动化的数据处理系统。通过实验验证, 该系统显著提高了数据处理的效率和准确性, 展示了在教育数据处理中的广泛应用潜力。

**关键词:** Openpyxl; 高校期末成绩; 数据处理; 自动化; Python

DOI:10. 69979/3029-2735. 24. 3. 032

## 1. 引言

### 1.1 研究背景

随着信息技术的飞速发展, 教育信息化已经成为现代教育的重要趋势之一。在高校教育管理中, 学生成绩数据的处理与管理一直是一个关键环节。传统的成绩管理方式主要依赖于人工录入和手动处理, 不仅耗时费力, 还容易出现错误。尤其在期末考试成绩汇总与分析过程中, 手工操作的效率和准确性难以满足当前高效、精准的需求。近年来, 自动化技术在教育管理中的应用逐渐增多, 特别是利用编程语言和数据处理工具进行成绩数据的自动化处理, 显著提升了工作效率和数据准确性。Python 作为一种高效、简洁且功能强大的编程语言, 在数据处理领域表现尤为出色。其丰富的第三方库提供了多种数据处理解决方案, 例如, Pandas、NumPy 和 Openpyxl, 可以方便地进行 Excel 数据的读取、清洗、转换和分析<sup>[1]</sup>, 其中 Openpyxl 库专门用于操作 Excel 文件, 因其易用性和强大功能受到广泛关注。

### 1.2 研究目的

本研究旨在探索基于 Python 和 Openpyxl 库的高校期末成绩数据处理方法, 通过自动化的方式实现成绩数据的读取、处理、计算和格式化, 以提高数据处理的效率和准确性。同时, 研究结果将为高校成绩管理提供一种高效、可靠的技术解决方案, 推动教育数据处理的自动化发展。具体来说, 本研究的目标包括: 设计并实现一个自动化的成绩数据处理系统; 提高成绩数据处理的效率和准确性; 为其他教育数据处理应用提供参考和借鉴。

### 1.3 研究意义

本研究具有重要的理论和实践意义。从理论上讲,

本研究丰富了教育数据处理领域的技术手段, 展示了 Python 和 Openpyxl 在实际应用中的潜力和优势。从实践上讲, 研究结果可以直接应用于高校的成绩管理系统, 解决实际工作中的数据处理难题, 减轻教师的工作负担, 提高工作效率和数据质量。此外, 本研究还具有一定的推广价值。通过对成绩数据处理的研究, 可以为其他类似的教育数据处理应用提供技术参考, 推动教育管理的信息化和智能化发展。

## 2. 相关研究

### 2.1 数据处理与分析技术的现状

在教育管理中, 数据处理与分析技术的应用越来越广泛。传统的成绩管理主要依赖手工操作和基本的电子表格工具, 如 Microsoft Excel。然而, 这些方法往往存在效率低、易出错的问题。随着大数据和人工智能技术的发展, 教育数据处理技术逐渐向自动化、智能化方向转变。现有的成绩管理系统如学信网、教务管理系统等, 虽然能够实现基本的数据处理功能, 但其灵活性和可扩展性有限, 难以满足复杂的数据分析需求方式<sup>[2]</sup>。

### 2.2 Openpyxl 库的应用

Openpyxl 是 Python 中一个强大的 Excel 文件处理库, 广泛应用于各类数据处理场景。与传统的 Excel 操作方式相比, Openpyxl 能够高效地进行数据读取、写入、格式化等操作, 极大地提高了数据处理的自动化程度。许多研究表明, Openpyxl 在教育、财务、统计等领域具有重要应用价值。例如, 马宁与陈曦等 (2020) 在其研究中指出, Openpyxl 库可操作 excel 数据表, 这些库可以极大地节约时间, 提升工作效率<sup>[3]</sup>。

## 3. 研究方法

### 3.1 研究设计

本研究为批量处理一批期末成绩数据，某一数据如图 1 所示，其余数据结构类似，该数据中各项权重为百分制，需要按要求修改为不同的比例，总权重为 100%。另外，考试成绩需要有具体的体现，考试成绩数据如图

2 所示，其中只需要保留成绩信息，并且最终还需计算综合成绩，并进行格式化处理等操作。最终处理结果如图 3 所示。

学生姓名	学号/工号	学校	院系	专业	班级	课程合视频(100%)	章节测验(100%)	作业(100%)	考试(100%)	签到(100%)	课程互动(100%)
王博	202200322532	...	...	...	233225	100.00	25.00	59.10	93.40	76.00	47.33
郝博	202200322640	...	...	...	233225	100.00	0.00	48.30	49.25	68.00	45.33
郝博	202200322646	...	...	...	233225	100.00	0.00	91.20	92.83	80.00	56.40
张博	202300022306	...	...	...	233225	46.15	75.00	87.10	91.75	84.00	53.60
张博	202300022320	...	...	...	233225	100.00	100.00	83.10	93.18	100.00	71.07
黄博	202300022407	...	...	...	233225	96.15	0.00	92.50	80.12	92.00	66.53
刘博	202300022418	...	...	...	233225	100.00	100.00	66.60	50.52	80.00	31.33
罗博	202300022421	...	...	...	233225	100.00	100.00	93.00	95.80	100.00	65.07
王博	202300022429	...	...	...	233225	100.00	100.00	59.60	93.15	96.00	61.67
赵博	202300022440	...	...	...	233225	34.60	0.00	59.00	89.75	88.00	49.33
曹博	202300322501	...	...	...	233225	100.00	100.00	85.00	94.70	92.00	65.33

图 1 处理前的数据

文件

开始

插入

公式

数据

审阅

视图

工具

会员专享

效率

WPS AI

帮助

撤销

粘贴

格式刷

字体

段落

背景

表格

数据

图表

窗口

帮助

公式

数据

表格

窗口

帮助

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

<

图 2 考试成绩数据

学生姓名	学号/工号	课程名称	课程成绩 (5%)	作业 (10%)	第一次测试	期中测试	期末考试	综合成绩 (40%)	课程成绩 (10%)	课程成绩 (15%)	最终成绩
王博	02200322532	20.00	1.25	5.91	79	96.97	23.67	37.36	7.60	7.10	79
郝明	02200322640	20.00	0.00	4.83	0	93.18	17.75	19.70	6.80	6.80	60
彭	02200322646	20.00	0.00	9.12	84	92.8	22.33	37.13	8.00	8.46	83
郭	02200322306	9.23	3.75	8.71	79	94.12	23	36.70	8.40	8.04	75
野	02200322320	20.00	5.00	8.31	84	93.94	22.33	37.27	10.00	10.66	91
曹	02200322407	19.23	0.00	9.25	73.8	71.59	20.92	32.05	9.20	9.98	80
刘	02200322418	20.00	5.00	6.66	0	81.44	21.75	20.21	8.00	4.70	64
廖	02200322421	20.00	5.00	9.30	86	94.7	23.58	38.32	10.00	9.76	92
王	02200322429	20.00	5.00	5.96	82	93.94	23.08	37.26	9.60	9.25	87
赵	02200322440	6.92	0.00	5.90	86	78.21	22.67	35.90	8.80	7.40	65
曹	02200322501	20.00	5.00	8.50	82	96.79	23.67	37.88	9.20	9.80	90
杨	02200322600	20.00	5.00	9.35	77.8	80.38	22.67	35.75	9.20	8.03	87

图 3 处理后的数据

本研究设计一个基于 Python 和 Openpyxl 的期末成绩数据处理系统。研究框架为:

1. 样本选择与数据来源: 选取高校期末考试成绩数据作为样本, 数据来源于与学校合作的学习通教学平台。
2. 数据处理流程: 包括文件管理、数据读取、数据处理、格式调整和结果输出五个步骤。
3. 数据分析方法: 采用统计分析和可视化方法, 对处理结果进行验证和分析。

### 3.2 数据处理流程

数据的处理过程如图所示, 主要包括如下四个步骤:

步骤一: 文件夹和文件管理

首先, 在用户桌面上创建文件夹, 存放所有待处理的成绩文件。每个文件对应一个班级的期末成绩数据, 文件为后缀是 xlsx 格式的 Excel 表格。

步骤二: 数据读取与预处理

使用 Openpyxl 库读取 Excel 文件, 并对数据进行预处理, 包括拆分合并单元格、清洗无效数据等。通过定义函数拆分合并单元格, 确保数据的完整性和可操作性。

步骤三: 数据处理

对预处理后的数据进行成绩计算与转换。例如, 将非百分制成绩转换为百分制, 按要求插入行列并进行合并、转换各项成绩比重, 并计算总评成绩等操作。同时, 确保数据结果的可读性和一致性。

步骤四: 数据格式化与保存

对处理后的数据进行格式化, 包括设置单元格边框、字体、对齐方式和背景填充色等。最终将处理后的数据保存为新文件。



图 4 数据处理流程

## 4. 实验与实现

### 4.1 环境搭建

在进行数据处理之前, 需要搭建一个适当的开发环境。本文选择使用 Python 编程语言, 主要工具包括 Python 3.x 和 Openpyxl 库。此外, 还需要安装其他辅助库, 如 os 和 shutil, 用于文件管理和操作。开发环境的配置步骤如下:

1. 安装 Python: 从 Python 官方网站下载并安装最新版本的 Python 3.x。
2. 安装 Openpyxl: 在命令行界面输入以下命令: `pip install openpyxl`
3. 安装其他辅助库: `pip install os ; pip install shutil`

### 4.2 代码实现

本节介绍具体的代码实现, 包括文件操作、数据读取与预处理、成绩计算与转换以及数据格式化与保存。

#### 4.2.1 文件操作

首先, 在用户桌面上创建一个名为“待处理成绩数据”的文件夹, 并将所有待处理的成绩文件存放其中。考虑到程序跨平台的兼容性, 通过使用 `os.path.expanduser('~')` 和 `os.path.join()`, 从而使代码可以在不同操作系统 (如 Windows、macOS 和 Linux) 上运行, 而无需手动调整路径格式。用户只需传递一个文件夹名, 代码会自动生成完整的路径, 避免了手动拼接路径时可能出现的错误。然后, 通过 python 自带的 os 库, 获取文件夹的路径, 同时通过自定义函数获取每个文件的路径, 便于下一步的数据读取。部分代码如图 5 所示:

```

def document_dir(dirs):
    # 获取文件夹路径
    user_dir = os.path.expanduser('~')
    doc_dir = os.path.join(user_dir, 'Desktop', dirs)
    return doc_dir
    
```

图 5 获取桌面文件夹路径代码

#### 4.2.2 数据读取与数据预处理

借助 Openpyxl 库读取 Excel 文件, 并对数据进行预处理, 清洗无效数据, 对源数据中的部分行列进行删除, 以及插入等操作。通过定义函数拆分合并单元格, 确保数据的完整性和可操作性。部分代码如图 6 所示:



```
def unmerge_cell(ws):
    """
    查找合并单元格，并拆分
    """
    m_list = ws.merged_cells # 合并单元格的位置信息
    cr = []
    for m_area in m_list:
        # 合并单元格的起始行坐标、终止行坐标。。。。,
        r1, r2, c1, c2 = m_area.min_row,
        m_area.max_row, m_area.min_col, m_area.max_col
        # 横向合并单元格的位置信息提取
        if c2 - c1 > 0 or r2 - r1 > 0:
            cr.append((r1, r2, c1, c2))
            print('符合条件%s' % str(m_area))
    for r in cr:
        ws.unmerge_cells(start_row=r[0], end_row=r[1],
            start_column=r[2], end_column=r[3])
```

图 6 查找合并单元格并拆分程序

#### 4.2.3 数据处理

在预处理后的数据基础上再次进行成绩数据的处理。这里主要进行的操作有：1、在考试统计表中获取多次考试的成绩，写入到对应的考试成绩详情列中。2、将课程音视频，章节测验等项目的权重由 100%按要求转化为特定的权重值。3、对考试成绩中非百分制的成绩需进行换算为百分制，4、计算最终的综合成绩并将综合成绩列的值进行四舍五入处理。部分代码如图 7 所示：

```
# 处理考试统计表
ws2 = wb['考试统计']
col_list = []
for j in range(1, ws2.max_column + 1):
    if ws2.cell(4, j).value == '成绩':
        col_list.append(j)
# 复制考试数据到成绩详情表
ws1.cell(1, 6).value = ws2.cell(1, col_list[0]).value
ws1.cell(1, 7).value = ws2.cell(1, col_list[1]).value
ws1.cell(1, 8).value = ws2.cell(1, col_list[2]).value
for r in range(2, ws1.max_row + 1):
    #成绩总分为100的情况
    ws1.cell(r, 6).value = round(float(ws2.cell(r, col_list[
0]).value)/1,2)
    #成绩总分为不为100的情况，如：250分、300分
    ws1.cell(r, 7).value = round(float(ws2.cell(r, col_list[
1]).value)/2.5,2)
    ws1.cell(r, 8).value = round(float(ws2.cell(r, col_list[
2]).value)/3.0,2)
# 将综合成绩列的值进行四舍五入
print(round(float(ws1.cell(1, 12).value)))
for i in range(3, ws1.max_row + 1):
    value_cj=float(ws1.cell(i, 12).value)
    ws1.cell(i, 12).value=str(round(value_cj))
```

图 7 部分成绩处理程序

#### 4.2.4 数据格式化与保存

成绩数据处理完成后，需对其进行格式化，包括表头和单元格中字体的样式，大小，对齐方式，单元行和列的宽度。借助 openpyxl 库的 styles 模块即可很好的实现。最好将格式化好的文件进行重命名并保存。部分代码如图 8 所示：

```
from openpyxl.styles import Border, Side, Alignment, Font,
PatternFill
# 设置样式
border1 = Border(left=Side(style='thin'), right=Side(
style='thin'), top=Side(style='thin'), bottom=Side(
style='thin'))
align = Alignment(horizontal='center', vertical='center')
font1 = Font(bold=True, color= FFFFFFFF )
font2 = Font(bold=False, color= 000000 )
fill_1 = PatternFill(fill_type='solid', fgColor= 000000 )
fill_2 = PatternFill(fill_type='solid', fgColor= DDDDDD )
# 数据格式化
for i in range(1, ws.max_row + 1):
    for j in range(1, ws.max_column + 1):
        ws.cell(i, j).border = border1
        ws.cell(i, j).alignment = align
        if i == 1 or i == 2:
            ws.cell(i, j).font = font1
            ws.cell(i, j).fill = fill_1
        else:
            ws.cell(i, j).font = font2
            if i % 2 == 0:
                ws.cell(i, j).fill = fill_2
# 保存处理后的文件
new_path = os.path.join(path, '处理后的成绩文件.xlsx')
wb.save(new_path)
```

图 8 数据格式化与保存程序

## 5. 结果与分析

在完成代码实现后，进行测试以验证数据处理的准确性。将处理结果与手工处理结果进行比较，确保数据处理的正确性和一致性。实验结果表明，基于 Python 和 OpenPyXL 的期末成绩数据处理系统显著提高了数据处理的效率和准确性。在测试过程中，系统能够快速、准确地读取和处理大量成绩数据，并生成符合要求、格式规范的结果文件。具体结果分析如下：

1. 数据读取与处理效率：系统能够在短时间内读取并处理大量 Excel 文件，显著减少了人工操作的时间。
2. 数据准确性：系统通过自动化处理，避免了人工录入和计算错误，提高了数据的准确性。
3. 数据格式规范性：处理后的成绩文件格式规范，便于后续的查阅和分析。
4. 可扩展性强：代码可扩展性强，易于根据实际需求进行修改和扩展。

## 6. 结论与展望

本研究通过设计并实现基于 Python 和 OpenPyXL 的期末成绩数据处理系统，验证了自动化技术在教育数据处理中的应用潜力。研究结果表明，自动化数据处理系统显著提高了数据处理的效率和准确性，解决了传统手工操作的诸多问题，为高校成绩管理提供了一种高效、可靠的技术解决方案。。

未来的研究可以进一步优化系统功能，增加可视化模块，实现更加智能化的教育数据处理和分析。此外，可以探索将该系统应用于其他教育管理领域，如学生考勤数据处理、教学质量评估等，以推动教育管理的信息化和智能化发展。

## 参考文献

- [1] 刘易. Python实现Excel文档转换到Word文档的自动化方法[J]. 电脑编程技巧与维护, 2024, 467(05): 47-49. DOI: 10. 16184/j. cnki. compmg. 2024. 05. 017.
- [2] 杜江舟. 高校教务管理系统信息化建设研究[J]. 科教文汇, 2023, 597(21): 24-26. DOI: 10. 16871/j. cnki. kjwh. 2023. 21. 004.
- [3] 马宁, 陈曦, 张李铭. 基于Selenium与Openpyxl的Web脚本自动化设计研究[J]. 电脑知识与技术, 2020, 16(01): 56-58+75. DOI: 10. 14004/j. cnki. ckt. 2020. 0024.