**任务五、实战演练：数据采集、数据清洗**

**一、数据采集**

数据采集，又称数据获取，是指将数据从数据源采集到可以支持大数据架构环境的过程。数据采集是数据挖掘、数据分析的一个环节，在数据处理过程中是非常基本的操作步骤（如下图所示），也是数据分析道路上的重中之重。再好的分析原理、建模算法，没有高质量的数据都是没有用的。数据采集的质量直接决定了后续的分析是否准确。

**任务目标：**从上交所XBRL教学专用仿真网站采集上市公司的报表数据，了解数据采集的Python代码，理解爬虫的基本原理和步骤。

**任务实现方式：**Python语言实现。

**任务流程：**1.利用代码编辑器，使用Python代码进行单企业数据采集；2.利用代码编辑器，使用Python代码进行多企业数据采集；3.利用代码编辑器，使用Python代码进行多企业多表数据采集。

**任务准备：**上交所XBRL实例文档：XBRL (可扩展商业报告语言，eXtensible Business Reporting Language)，是 XML (可扩展的标记语言Extensible Markup Lanquage) 于财务报告信息交换的一种应用，是目前应用于非结构化信息处理尤其是财务信息处理的有效技术。XBRL技术在资本市场信息披露中的应用，使上市公司、监管机构、交易所、会计师事务所、投资者、研究机构证券信息服务商等信息加工者与使用者能够以更低的成本、更高的效率实现信息交换和共享，有效提高了信息披露透明度和监管水平，促进了资本市场的健康有序发展。教学专用XBRL仿真网站（如下图2-1）：在使用爬虫进行数据爬取时，有的网站有反爬机,制，有的网站会有监测，如果同时有数百或数千名同学使用同一IP段或IP同时访问上交所网站，该网站会检测到此IP异常，会暂时封闭此IP的访问权限，导致相关页面无法访问。基于此，任务研发了教学专用上交所XBRL仿真网站，该网站可以支持多人同时进行报表数据采集。



图2-1 教学专用XBRL仿真网站

**任务1：查看上交所XBRL网页内容**

进入上交所XBRL实例文档查看数据采集数据。

**操作步骤：**

1.单击开始任务，进入上交所网页，如图2-2所示

2. 查看列表页面；

3. 单击任意财报，进入财报详情页面；

4. 单击上方的财报类型筛选框，选择对应的财报季度和年份后确定；

5. 单击切换基本信息表、利润表等表格。



图2-2 教学专用XBRL仿真网站财报详情页面

**任务2：单企业数据采集**

利用大数据技术，在XBRL教学网站采集浦发银行2019年1季度基本信息。

股票代码：600000

采集数据年份:2019年

采集报表类型:1季度

采集数据内容:基本信息

**操作步骤：**

1.打开新道代码编译器编写代码

2.导入requests、pandas两个Python库文件代码如下：

# 一、导入Python库文件

import requests

import pandas as pd

3.输入浦发银行对应的股票代码和股票简称输入code变量中， 输入采集的报表年份在year变量中，输入采集的报表类型在repport\_period\_id变量中，输入请求连接在url变量中，代码如下：

# 二、请输入股票代码、报表年份、报表类型、请求连接

code = [("600000", "浦发银行")]

year = ["2019"]

report\_period\_id = ["4000"]

url=["https://ssecurity.seentao.com/debug/security/security.info.get"]

4.使用pandas库文件的缩写pd打开文件“中英指标对照”，将两列数据转换为字典类型，对照翻译指标名称，代码如下：

# 三、使用pandas库文件打开文件

name\_file = pd.read\_excel('数据采集/中英文对照表/中英指标对照.xlsx', sheet\_name='sz\_info')

name\_dir = dict(zip(name\_file['en'], name\_file['ch']))

5.使用pandas库建立一个DataFrame表，便于存储采集到的数据,代码如下：

data = pd.DataFrame()

6. 利用try ... Except ... 逻辑对数据采集过程进行处理，利用 for 循环逻辑，对该类数据进行采集，并保存采集到的数据信息，代码如下：

# 四、使用for循环采集数据

try:

for S\_id in code:

stock\_id = S\_id[0]

for b in report\_period\_id:

try:

postdata = {"stockId": stock\_id, "reporttype": b, "callType": "collection"}

json\_data = requests.post(url[0], postdata).json()['result']

df = pd.DataFrame(json\_data)

df = df[df['reportyear'].isin(year)]

data = pd.concat([data,df])

except Exception as e:

print(e)

continue

data.rename(columns=name\_dir, inplace=True)

print(data)

end\_data = data[name\_file['ch']]

end\_data.to\_excel('浦发银行2019年1季度基本信息数据采集结果.xlsx', encoding='utf-8', index=False)

except Exception as e:

print('采集失败', e)

Python代码运行如图2-3所示，运行结果见资料包-浦发银行2019年1季度基本信息数据采集结果文件。

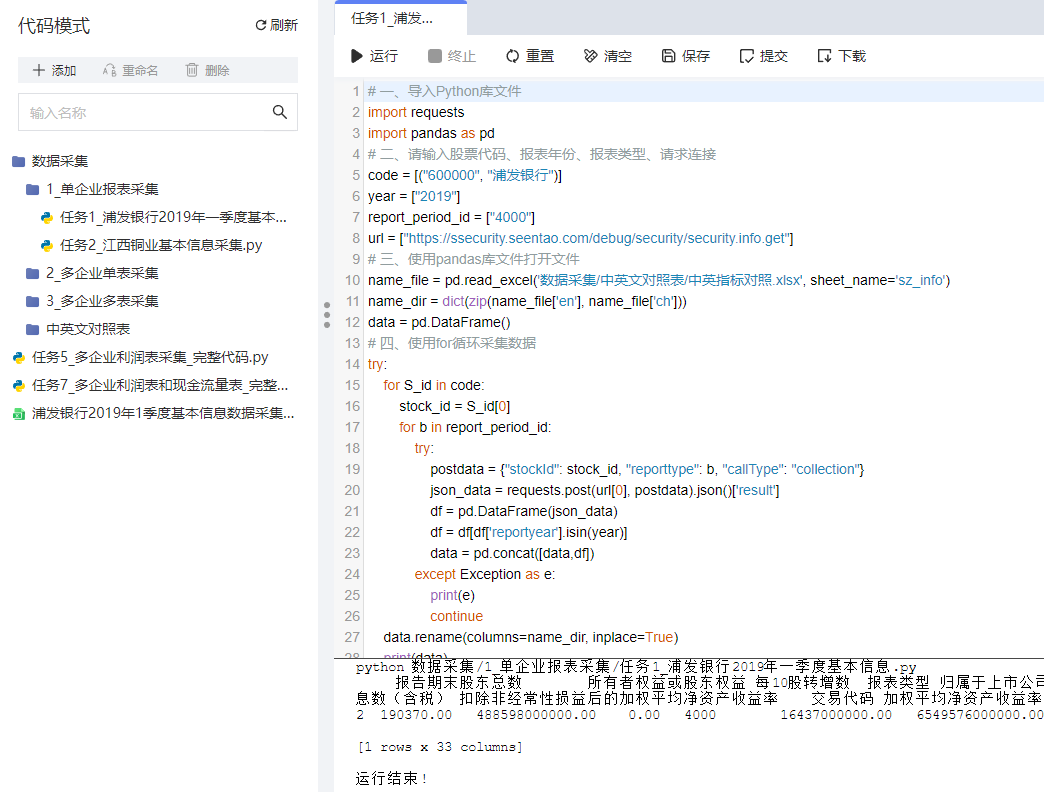


图2-3 浦发银行基本信息采集python运行

**任务3：多企业数据采集**

利用大数据技术，采集浦发银行与招商银行2015-2016年资产负债表数据。

企业简称:浦发银行、招商银行

股票代码:600000、600036

采集数据年份:2015年、2016年

采集报表类型:年报

采集数据内容:资产负债表、利润表

**操作步骤：**

1.打开新道代码编译器编写代码；

2.导入requests、pandas两个Python库文件，代码如下：

# 一、导入Python库文件

import requests

import pandas as pd

3.输入浦发银行和招商银行对应的股票代码和股票简称输入code变量中，输入采集的报表年份在year变量中，输入采集的报表类型在repport\_period\_id变量中，输入请求连接在url变量中，代码如下：

# 二、请输入股票代码、报表年份、报表类型、请求连接

code = [("600000", "浦发银行"),("600036", "招商银行")]

year = ["2015","2016"]

report\_period\_id = ["5000"]

url=["https://ssecurity.seentao.com/debug/security/security.balancesheet.get"]

4.使用pandas库文件的缩写pd打开文件“中英指标对照”，将两列数据转换为字典类型，对照翻译指标名称，代码如下：

# 三、使用pandas库文件打开文件

name\_file = pd.read\_excel('数据采集/中英文对照表/中英指标对照.xlsx', sheet\_name='sz\_balancesheet')

name\_dir = dict(zip(name\_file['en'], name\_file['ch']))

5.使用pandas库建立一个DataFrame表，便于存储采集到的数据,代码如下：

data = pd.DataFrame()

6. 利用try ... Except ... 逻辑对数据采集过程进行处理，利用 for 循环逻辑，对该类数据进行采集，并保存采集到的数据信息，代码如下：

# 四、使用for循环采集数据

try:

for S\_id in code:

stock\_id = S\_id[0]

for b in report\_period\_id:

try:

postdata = {"stockId": stock\_id, "reporttype": b, "callType": "collection"}

json\_data = requests.post(url[0], postdata).json()['result']

df = pd.DataFrame(json\_data)

df = df[df['reportyear'].isin(year)]

data = pd.concat([data,df])

except Exception as e:

print(e)

continue

data.rename(columns=name\_dir, inplace=True)

print(data)

end\_data = data[name\_file['ch']]

end\_data.to\_excel('浦发银行&招商银行2015-2016年资产负债表采集结果.xlsx', encoding='utf-8', index=False)

except Exception as e:

print('采集失败', e)

Python代码运行如图2-4所示，运行结果见资料包-浦发银行&招商银行2015-2016年资产负债表采集结果文件。

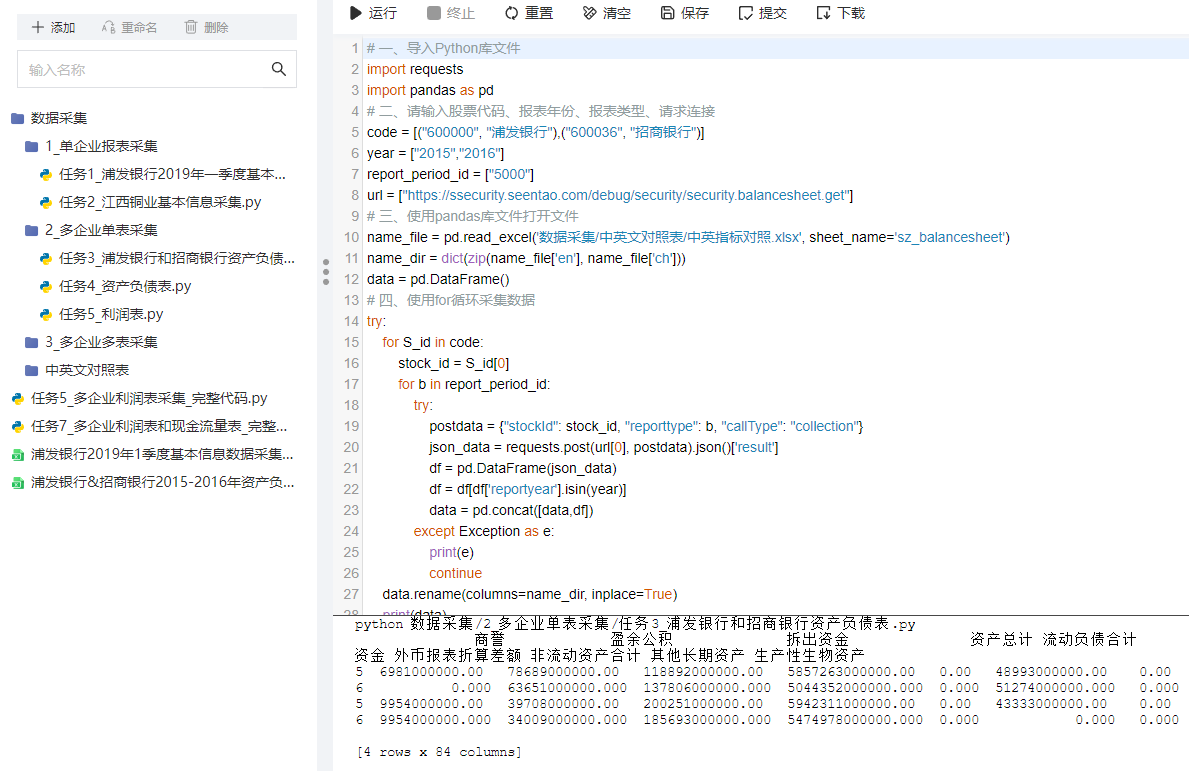


图2-4 浦发银行与招商银行资产负债表采集代码运行

**任务4：多企业多报表数据采集**

利用大数据技术，采集浦发银行与招商银行2015-2016年资产负债表和利润表数据。

企业简称:浦发银行、招商银行

股票代码: 600000、600036

采集数据年份:2015年、2016年

采集报表类型:年报

采集数据内容:资产负债表、利润表

**操作步骤：**

1.打开新道代码编译器编写代码；

2.导入以下Python库文件：requests、pandas、re、time，代码如下：

# 一、导入Python库文件

import requests

import re

import pandas as pd

import time

2. 输入浦发银行和招商银行对应的股票代码和股票简称输入code变量中，输入采集的报表年份在year变量中，输入采集的报表类型在repport\_period\_id变量中，输入请求连接在url变量中，代码如下：

# 二、请输入股票代码、报表年份、报表类型、请求连接

code = [("600000", "浦发银行"), ("600036", "招商银行")]

year = ["2015", "2016"]

report\_period\_id = ["5000"]

url=["https://ssecurity.seentao.com/debug/security/security.balancesheet.get","https://ssecurity.seentao.com/debug/security/security.incomestatement.get"]

4.配置输出数据文件名翻译对照字典：balancesheet 对应资产负债表，配置输出数据文件名翻译对照字典：cashflow对应现金流量表，配置输出数据文件名翻译对照字典：incomestatement 对应利润表，代码如下：

# 三、配置输出数据文件名翻译对照字典

name\_dict = {"incomestatement": "利润表", "cashflow": "现金流量表", "balancesheet": "资产负债表"}

5.利用try ... Except ... 逻辑对数据采集过程进行处理，利用 for 循环逻辑，对该类数据进行采集，并保存采集到的数据信息，代码如下：

# 四、使用for循环采集数据

try:

for one in url:

name = re.search('\/security\.(.\*)\.get', one).group(1)

sheet\_name = 'sz\_' + name

data = pd.DataFrame()

for S\_id in code:

stock\_id = S\_id[0]

for b in report\_period\_id:

try:

postdata = {"stockId": stock\_id, "reporttype": b, "callType": "collection"}

json\_data = requests.post(one, postdata).json()['result']

df = pd.DataFrame(json\_data)

df = df[df['reportyear'].isin(year)]

data = pd.concat([data,df])

except Exception as e:

print(e)

continue

name\_file = pd.read\_excel('数据采集/中英文对照表/中英指标对照.xlsx', sheet\_name=sheet\_name)

name\_dir = dict(zip(name\_file['en'], name\_file['ch']))

data.rename(columns=name\_dir, inplace=True)

print(data)

end\_data = data[name\_file['ch']]

end\_data.to\_excel(f"{'&'.join([i[1] for i in code])}\_{name\_dict.get(name)}采集结果{int(time.time())}.xlsx",

encoding='utf-8', index=False)

except Exception as e:

print('采集失败', e)

Python代码运行如图2-5所示，运行结果见资料包-浦发银行&招商银行\_利润表采集结果，浦发银行&招商银行\_资产负债表采集结果文件。

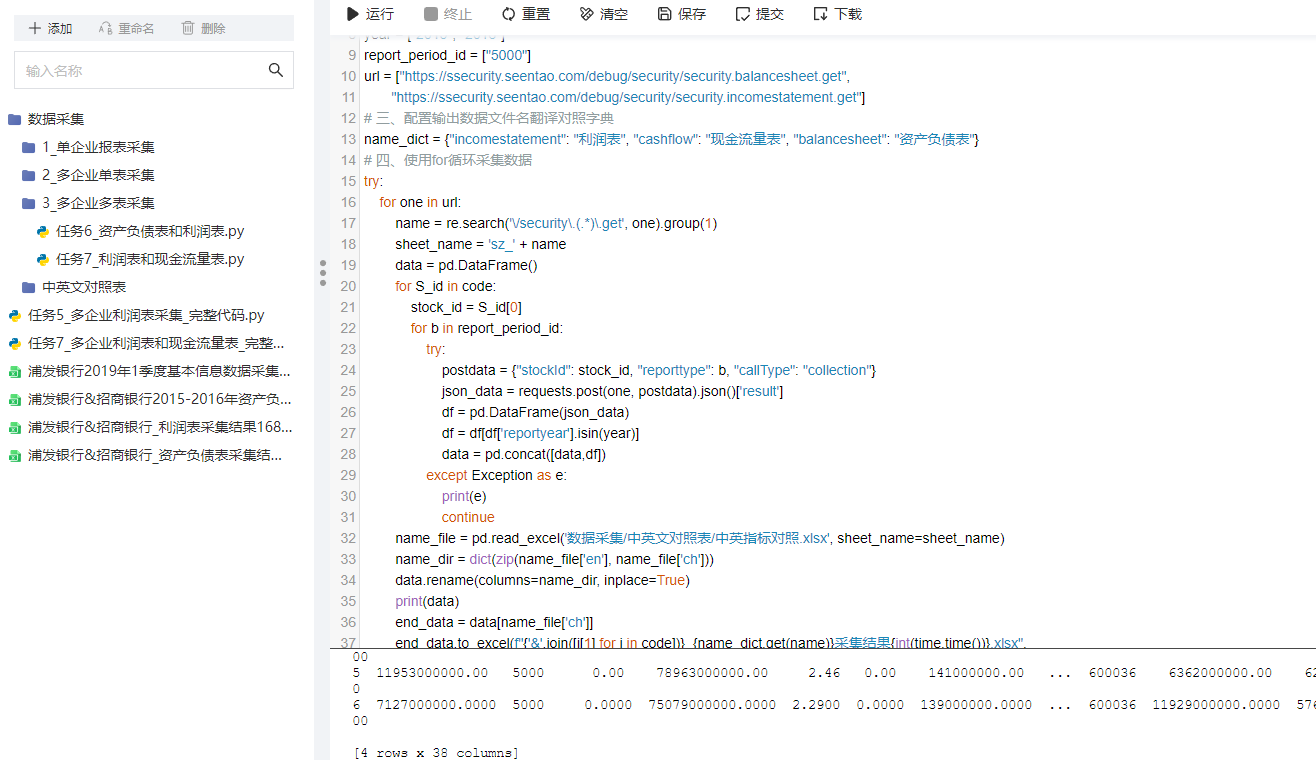


图2-5 浦发银行与招商银行资产负债表采集代码运行

**拓展任务：东方财富研报数据采集**

新高曲证券股份有限公司证券投资部投资决策需要有自己的研究报告做支撑，研究员小赵需要阅读大量研报，撰写出高质量的研报。如何使用爬虫技术批量获取中国核电(601985)的研报呢?

操作步骤：

1. 结合目标网站，确定需要抓取数据（如下图2-6）
2. 确定获取所需数据的方法及必要参数。
3. 编写代码，实现获取数据，处理数据，存储数据。



图2-6 东方财富网首页

**二、数据清洗**

B公司是一家销售办公用品、办公家具和办公电子设备的公司，旗下有多家直营店，每月月底，各直营店都会向财务部提供本月的订单详情表。现在公司的财务分析师手上有一份汇总多年的订单详情表。

财务分析师需要对汇总订单详情表进行分析，如客户群体分布分析、受欢迎商品分析、收入利润情况分析。在做分析之前，财务分析师需要对原始数据表进行数据清洗，构建一张准确的数据表是数据分析最重要的环节之一观察此表，我们可以发现，表中有一些空值和“\_”，有的单元格还有特殊字符，如下图2-7所示。对客户ID、产品名称这些数据也需要进一步数据处理，才可以准确的进行分析。

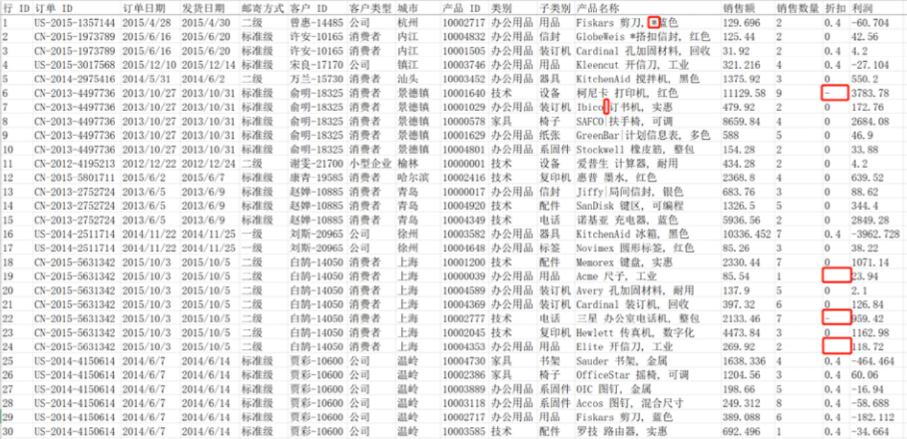


图2-7 原始数据表数据图

**任务目标：**对资源中的表格数据进行处理，进行特殊字符清洗、格式内容清洗、python数据清洗，了解数据清洗的概念、内容、方法及python数据清洗上。

**任务实现方式：**数据清洗工具与Python语言实现。

**任务流程：**1.对表中空值与“\_”字符进行清洗；2. 将表中的“客户ID”拆分为两列，分别为“客户名称”和“客户ID”；3.将“产品名称”中含有不该存在的字符，如“”“” “”“\*”替换为空；将“产品名称”拆分为三列，分别为“产品品牌”“产品品名”、“产品规格”三列。

**任务准备：**查看表格数据（如下图2-8），数据表含有以下信息:订单ID、订单日期发货日期、邮寄方式、客户ID、客户类型城市、产品ID、类别、子类别、产品名称、销售额、销售数量、折扣、利润。表内问题分析:1.数据中没有客户名称列字段。客户名称与客户ID混在“客户ID”字段中；2.数据表中没有产品名称列2字段，而是将产品品牌、产品名称、产品规格三个信息统一记录在“产品名称”字段中；3.表内有缺失值、有不该存在的字符。资料见资料包-销售数据\_清洗前文件。

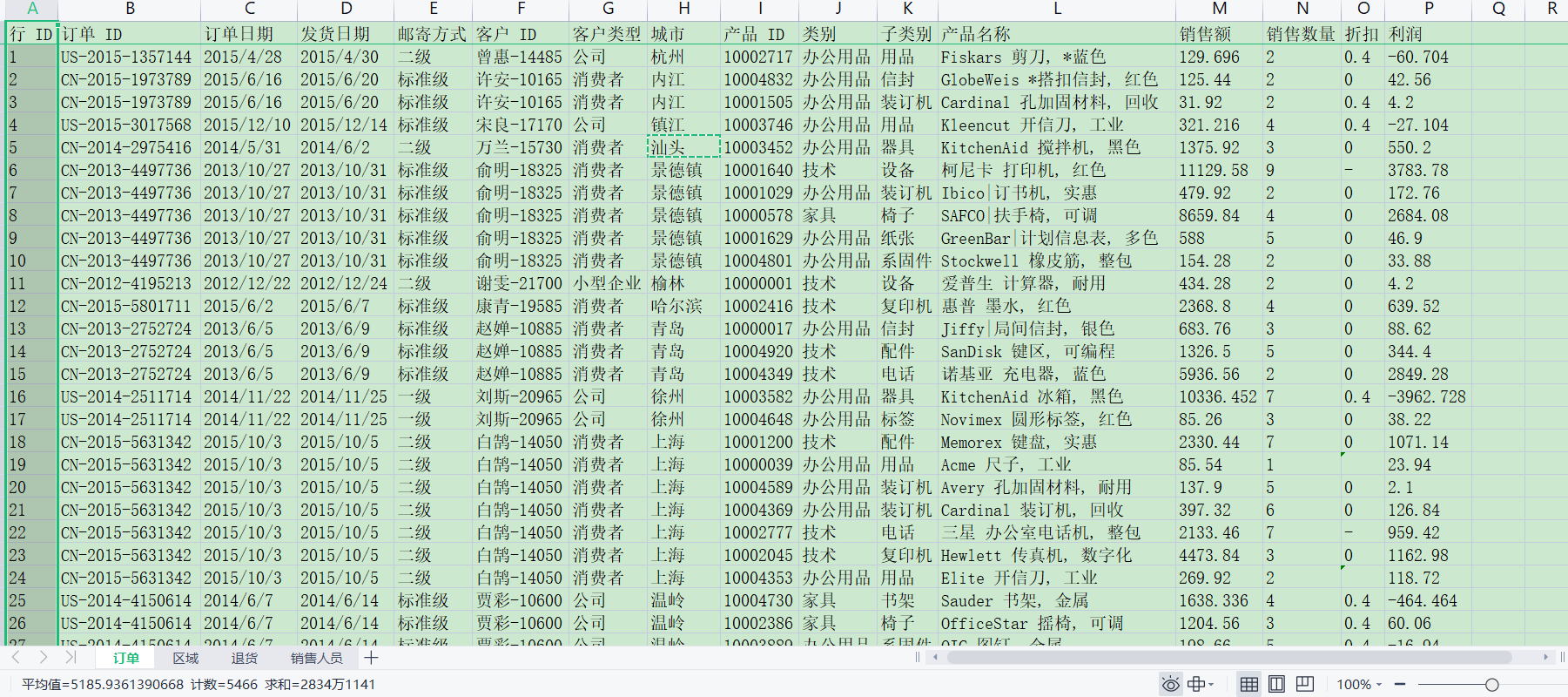


图2-8 清洗前表格数据图

**任务1：全局清洗**

特殊字符清洗，使用全局清洗规则对整张表的数据空格、“-”进行清洗，操作步骤工具如图2-9所示：

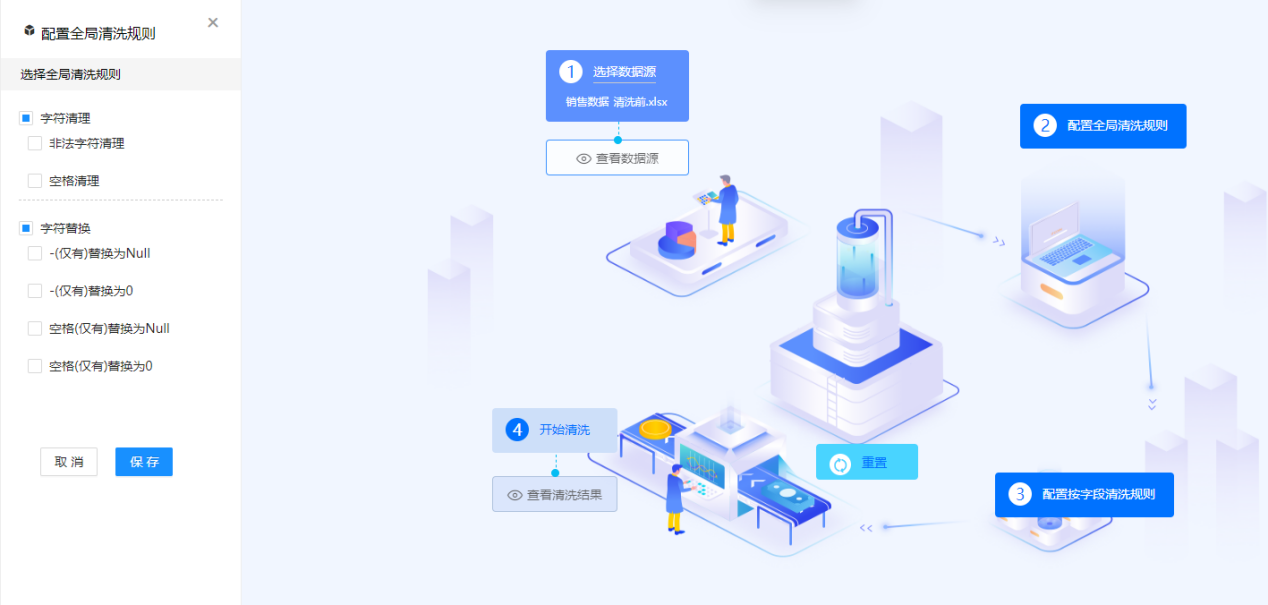
****

图2-9 全局清洗工具界面

**1.清洗空格**

**操作步骤：**

（1）选择数据源

①单击【选择数据源】，在下拉列表中找到预置的数据，单击【保存】；

② 如果下拉列表中没有预支数据，可以前往资源下载处，先下载本任务所需数据，单击【上传数据】，选择在资源下载处下载的数据，单击【保存】；

③平台会提示“保存成功”。

（2）配置清洗规则

①单击【配置全局清洗规则】；

②勾选要使用的规则，如“空格(仅有)替换为0”，单击【保存】；

③平台会提示“配置清洗规则成功”。

（3）开始清洗

①单击[开始清洗]；

②单击【确定】；

③系统开始清洗数据，会出现一个“清洗中...”的模型；

④清洗完毕，界面显示“清洗并上传成功”。

（4）：下载数据并保存到本地电脑，运行结果见资料包-清洗任务1操作结果文件。

**2.清洗“-”**

**操作步骤：**

（1）选择数据源

①将任务1操作结果上传到数据清洗系统中，单击【保存】；

②平台会提示“保存成功”。

（2）配置清洗规则

①单击【配置全局清洗规则】；

②选择勾选要使用的规则，如 “-(仅有)替换为0” ，单击【保存】；

③平台会提示“配置清洗规则成功”。

（3）数据清洗

①单击【开始清洗】；

②平台提示“清洗中”，清洗结束后，平台会提示“清洗并上传成功”；

③单击【查看清洗结果】，对清洗结果进行阅览。

（4）下载数据并保存到本地电脑，运行结果见资料包-清洗任务2操作结果文件。

任务作业：请将全局清洗规则中特殊字符清洗结果提交。

**任务2：按字段清洗**

特殊字符清洗，将数据表中的“\*”清洗掉，将数据表中产品名称字段中的"/""\""|"清洗掉。格式内容清洗将数据表中客户ID字段中的客户名称与客户ID进行切分；将数据表中产品名称字段进行清洗，切分为“产品品牌”与“品名与规格”；将数据表中品名与规格字段进行清洗，切分为“产品品名”与“品名规格”。操作步骤工具如图2-10所示，在开始操作之前，需要单击【重置】，将数据清洗系统重置初始状态。

****

图2-10 按字段清洗工具界面

**1.产品名称清洗“\*”**

**操作步骤：**

（1）选择数据源

① 单击【上传数据】；

②将 任务2 操作结果上传，单击【保存】；

（2）配置清洗规则

①单击【配置按字段清洗规则】，

②单击【添加规则】，选择“字符替换”清洗规则；

③单击规则下方的【+】，选择需要切分的字段，在弹出的窗口中选择“产品名称”，并单击右移（添加）按钮，单击【确定】；

④ 填写“原内容”与“替换内容”，将“产品名称”中的“ \* ”替换为空，单击【保存】；

（3）数据清洗

①单击【开始清洗】；

②查看清洗结果；

（4）下载清洗数据并将数据保存至本地，运行结果见资料包-清洗任务3操作结果文件。

**2.产品名称清洗"/"、"\"、"|"**

**操作步骤：**

（1）选择数据源

①单击【上传数据】；

② 将 任务3 操作结果上传，单击【保存】；

（2）配置清洗规则

①单击【配置按字段清洗规则】，

②单击【添加规则】，选择“字符替换”清洗规则；

③单击规则下方的【+】，选择需要切分的字段，在弹出的窗口中选择“产品名称”，并单击右移（添加）按钮，单击【确定】；

④填写“原内容”与“替换内容”，

（3）数据清洗

①单击【开始清洗】；

② 查看清洗结果；

（4）下载清洗数据并将数据保存至本地，运行结果见资料包-清洗任务4操作结果文件。

**3.客户ID清洗**

**操作步骤：**

（1）选择数据源

①单击【上传数据】；

② 将 任务4 操作结果上传，单击【保存】；

（2）配置清洗规则

①单击【配置按字段清洗规则】，

② 单击【添加规则】，选择“字段切分”清洗规则；

③单击规则下方的【+】，选择需要切分的字段，在弹出的窗口中选择“客户ID”，并单击右移（添加）按钮，单击【确定】；

④ 将“客户ID”进行字段切分，切分分隔符为“-”，切分后名称为“客户名称”与“客户ID”；

⑤ 单击【保存】；

（3）数据清洗

①单击【开始清洗】；

①查看清洗结果；

（4）下载清洗数据并将数据保存至本地，运行结果见资料包-清洗任务5操作结果文件。

**任务4：Python数据清洗**

特殊字符清洗，利用大数据技术清洗数据表中的“空格”、“-”；

格式内容清洗，针对数据表中以下字段进行数据清洗：将客户ID字段进行清洗；将产品名称字段进行清洗；利用大数据技术，将财务报表中的数据进行数据类型清洗。

非需求性数据清洗，利用大数据技术，将无用数据进行清洗。

**1.“空格”与“-”清洗**

**任务资源：**

资料见资料包-销售数据\_清洗前文件。

**操作步骤：**

（1）打开新道代码编译器

（2）导入Python库文件，代码如下：

import pandas as pd

（3）定位需要清洗的文件路径，代码如下：

file\_name = '数据清洗/Python清洗/01\_任务/销售数据\_清洗前.xlsx'

（4）用Python库文件阅读数据清洗表，代码如下：

df = pd.read\_excel(file\_name)

（5）编写清洗规则，代码如下：

df2 = df.replace("-", 0)

df2 = df2.replace(" ", 0)

（6）保存清洗之后的文档，代码如下：

df2.to\_excel('任务6清洗结果.xlsx', index=False, encoding='utf-8-sig')

Python“空格”与“-”清洗代码运行如图2-11所示，运行结果见资料包-任务6清洗结果文件。



图2-11 空格与“-”清洗代码运行

**2.客户ID与产品名称清洗**

**操作步骤：**

（1）打开新道代码编译器

（2）导入Python库文件，代码如下：

import pandas as pd

（3）定位需要清洗的文件路径，代码如下：

file\_name = '任务6清洗结果.xlsx'

df = pd.read\_excel(file\_name)

（4）读取数据，代码如下：

df = pd.read\_excel(file\_name)

（5）编写清洗规则：数据类型转换，代码如下；

df[['客户名称', '客户 ID']] = df['客户 ID'].str.split('-', n=1, expand=True)

df[['产品品牌', '品名与规格']] = df['产品名称'].str.split(' ', n=1, expand=True)

df.drop(columns='产品名称', inplace=True)

df['品名与规格'].replace(' ', '', regex=True, inplace=True)

df[['产品品名', '产品规格']] = df['品名与规格'].str.split(',', n=1, expand=True)

df.drop(columns='品名与规格', inplace=True)

（6）将清洗结果进行保存

df.to\_excel('任务7清洗结果.xlsx', index=False, encoding='utf-8-sig')

客户ID与产品名称清洗代码运行如图2-12所示，运行结果见资料包-任务7清洗结果文件。

****

图2-12 客户名称与ID python代码运行

**3.无用数据清洗**

**任务资源：**

资料见资料包-华扬联众\_利润表\_无用数据清洗文件。

**操作步骤：**

（1）打开新道代码编译器

（2）导入Python库文件，代码如下：

# -\*- coding:utf-8 -\*-

import pandas as pd

（3）定位需要清洗的文件路径,代码如下：

# 指定数据源文件

file\_name = '数据清洗/Python清洗/06\_任务8/华扬联众\_利润表\_无用数据清洗.xlsx'

（4）构建列表，代码如下：

# 不需要内容列表 （后续数据处理中不需要的字段，将这些不需要的字段提前删除）

ignore\_list = ['金融资产利息收入', '已赚保费', '手续费及佣金收入',

'金融资产利息支出', '手续费及佣金支出', '退保金', '赔付支出净额','提取保险合同准备金净额', '保单红利支出', '分保费用']

（5）读取数据，代码如下：

# 读取数据

df = pd.read\_excel(file\_name, sheet\_name='报表数据')

（6）删除不需要的数据，代码如下：

df.drop(columns=ignore\_list, inplace=True)

（7）将清洗结果进行保存，代码如下：

# 保存处理后数据

df.to\_excel('任务8清洗结果.xlsx', encoding='utf-8', index=False)

Python代码运行如图2-13所示，运行结果见资料包-任务8清洗结果文件。

****

图2-13 华扬联众\_利润表\_无用数据清洗代码运行

**任务作业：**

请将关联性验证数据清洗结果提交

资料见资料包-员工薪酬数据\_清洗前文件，按清洗要求完成清洗任务。

要求：1.员工薪酬数据特殊字符与空格清洗2.员工薪酬数据字符替换3.员工薪酬数据均值填补与字段切分