利用Python爬取汽车之家指定数据

1.介绍项目

2.分析网站结构

3.说明技术选择

4.excel表格构建字段

5.Python如何连接excel表格

需求：

1.获取口碑频道SUV紧凑型车全部的数据

2.车名，价格，特点，评分

https://k.autohome.com.cn/#pvareaid=6848948

学员提供：

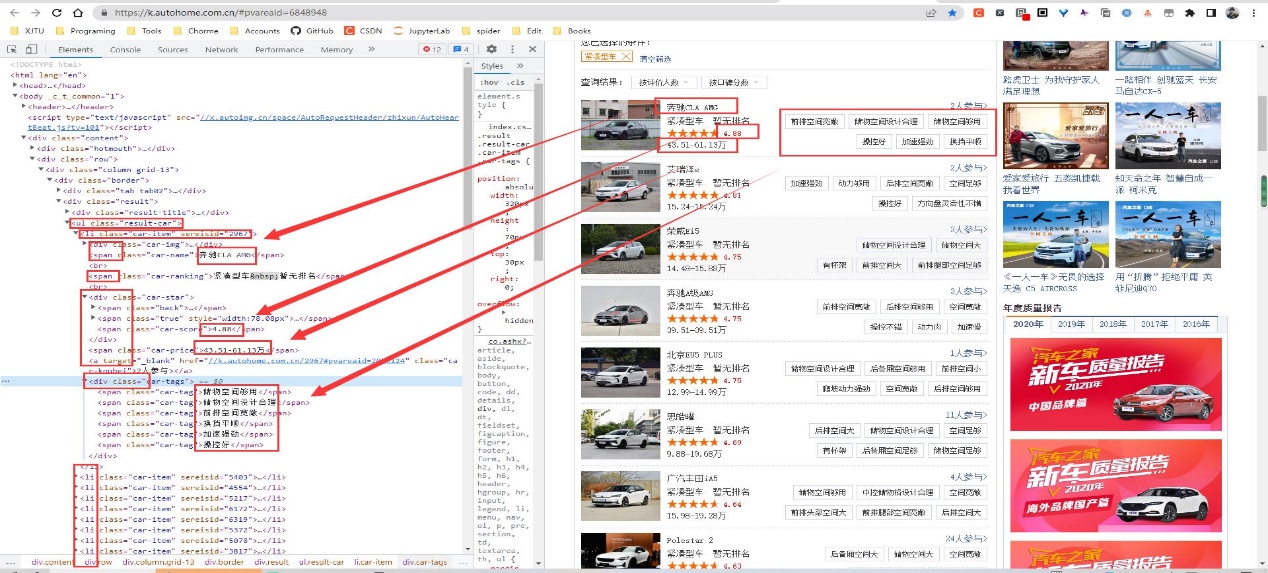
1. 代码(注释)

2. 文章(说明自己爬取的思路和过程，图文并茂，docx文档)

积分：500

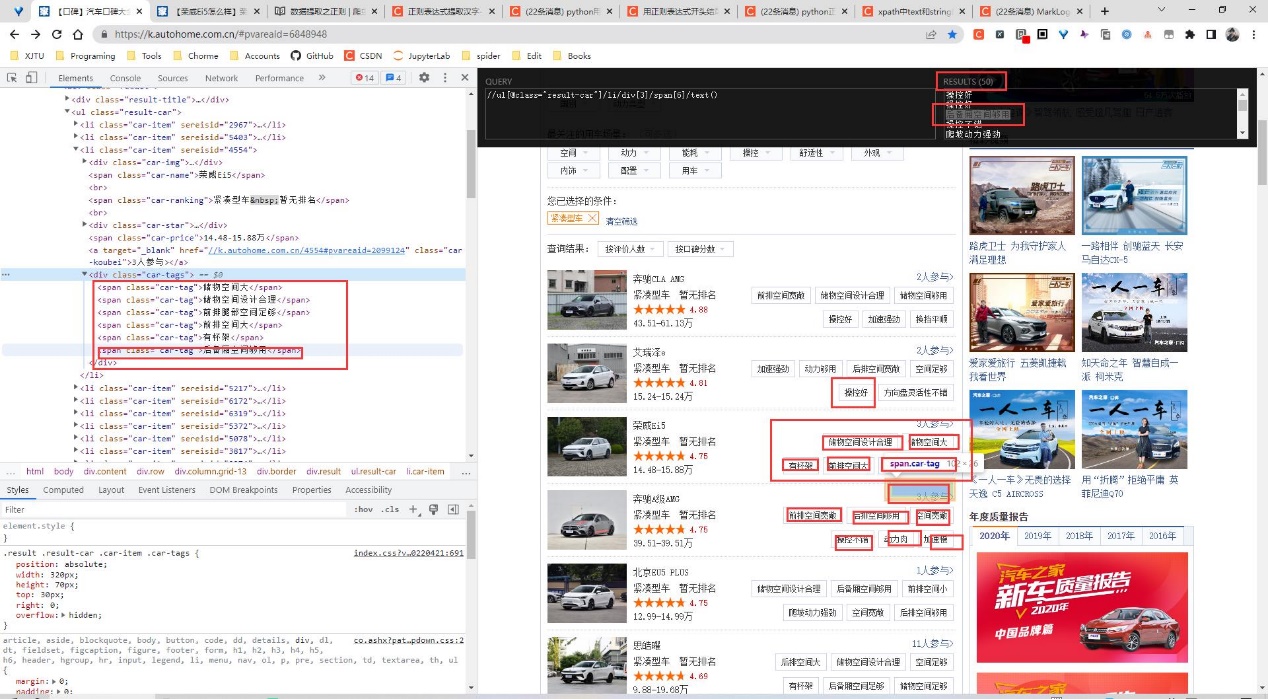
1. 网页分析

首先打开网页，发现在网页源代码中所有数据都可以找到，但是在框架文件中没有这些数据，所以可以通过selenium直接获取网页源代码，从而获取全部数据。



另外在初始页面中评价6有时会被隐藏不显示，如下图，但是在网页源码中是存在第6条评论数据的，所以爬取时对所有第六条数据都进行了爬取，如下是对4组数据的xpath语句。

'//li[@class="car-item"]/span[1]/text()' # # 车名的获取  
'//div[@class="car-star"]/span[3]/text()' # 评分的获取  
'//span[@class="car-price"]/text()' # 价格的获取  
  
'//div[@class="car-tags"]/span[1]/text()' # 评论1的获取  
'//div[@class="car-tags"]/span[2]/text()' # 评论2的获取  
'//div[@class="car-tags"]/span[3]/text()' # 评论3的获取  
'//div[@class="car-tags"]/span[4]/text()' # 评论4的获取  
'//div[@class="car-tags"]/span[5]/text()' # 评论5的获取  
'//div[@class="car-tags"]/span[6]/text()') # 评论6的获取



另外在初始页面中，并不是所有数据都直接显示出来的还有一部分需要点击加载更多才能加载出来。在selenium中可直接通过ID定位元素然后连续点击4次即可加载出所有数据，一共有50条数据。

click\_obj = chrome\_obj.find\_element*(*By.ID, "sceneGetMore"*)* # 定位 下页  
click\_obj.click*()* # 点击



其次就是Excel表格构建字段，按照如下代码即可。

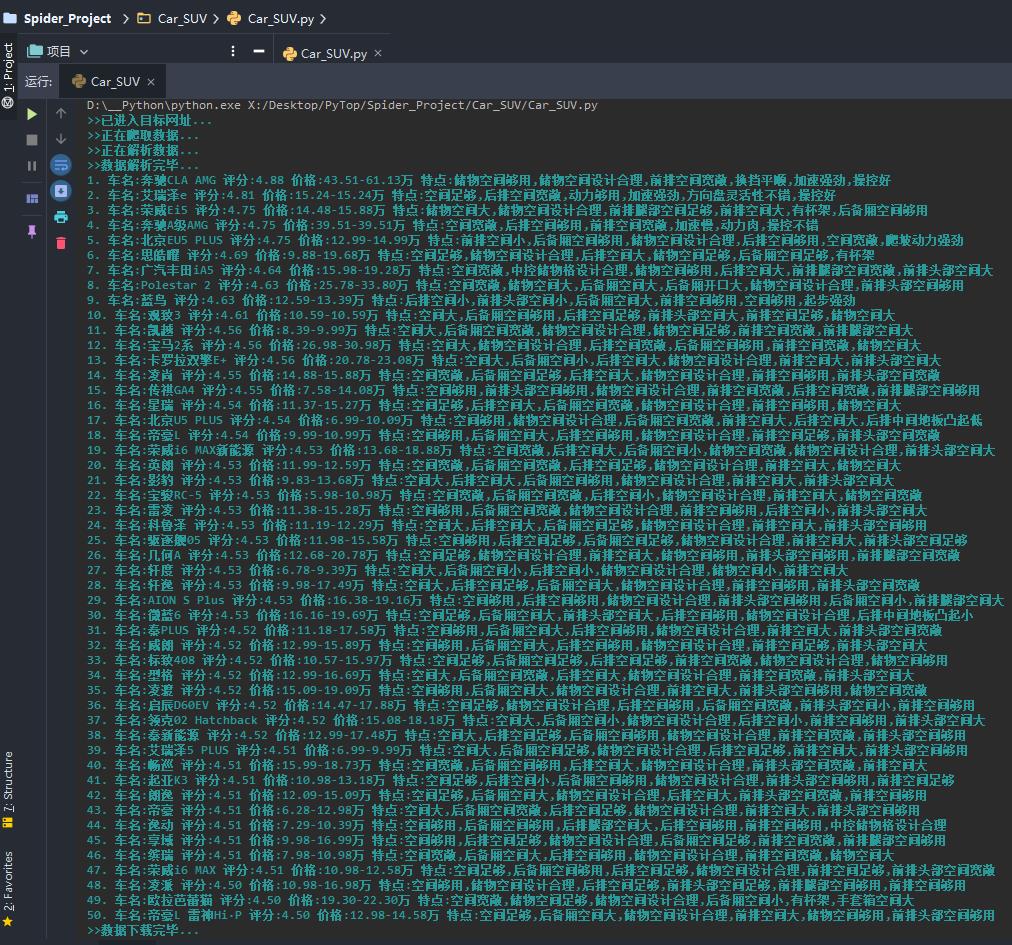
book = xlwt.Workbook*(*encoding='utf-8', style\_compression=0*)*sheet = book.add\_sheet*(*'SUV紧凑型车', cell\_overwrite\_ok=True*)*col = *(*"车名", "评分", "价格", "特点"*)*for j in range*(*len*(*col*))*:  
 sheet.write*(*0, j, col*[*j*])* # 构建表头  
 for i in range*(*1, len*(*car\_*)* + 1*)*: # 一列一列写入数据  
 sheet.write*(*i, 0, car\_*[*i - 1*])* # 车名  
 sheet.write*(*i, 1, score\_*[*i - 1*])* # 评分  
 sheet.write*(*i, 2, price\_*[*i - 1*])* # 价格  
 sheet.write*(*i, 3, tags\_*[*i - 1*])* # 特点  
book.save*(*'SUV紧凑型车.xls'*)*

1. 代码

*"""  
利用Python爬取汽车之家指定数据  
1.介绍项目  
2.分析网站结构  
3.说明技术选择  
4.excel表格构建字段  
5.Python如何连接excel表格  
需求：  
1.获取口碑频道SUV紧凑型车全部的数据  
2.车名，价格，特点，评分  
https://k.autohome.com.cn/#pvareaid=6848948  
学员提供：  
1. 代码(注释)  
2. 文章(说明自己爬取的思路和过程，图文并茂，docx文档)  
积分：500  
"""*import random  
from selenium import webdriver  
import time # 计时  
from lxml import etree # 解析html  
from selenium.webdriver.chrome.options import Options # 无头模式  
from selenium.webdriver.common.by import By  
import xlwt  
  
  
class SUV:  
 def \_\_init\_\_*(*self*)*:  
 self.Page = 4 # 爬取 新闻页数 15 条新闻/页  
 self.Error\_Num = 0 # 累计新闻正文下载失败个数  
 self.Success\_Num = 0 # 累计下载成功的新闻数  
 self.url = "https://k.autohome.com.cn/#pvareaid=6848948"  
 pass  
  
 def chrome\_obj*(*self*)*:  
 *"""创建一个Chrome对象并进入主页"""* options\_ = Options*()* # 设置无界面模式  
 options\_.add\_argument*(*'--headless'*)* chrome\_obj = webdriver.Chrome*(*options=options\_*)* # 无界面模式  
 chrome\_obj.get*(*self.url*)* print*(*"\033[1;36m>>已进入目标网址...\033[0m"*)* return chrome\_obj  
  
 def get\_page\_source*(*self, chrome\_obj*)*:  
 *"""获取网页源码"""* item\_page = chrome\_obj.page\_source  
 chrome\_obj.quit*()* # 结束 关闭 虚拟 Chrome  
 html\_obj = etree.HTML*(*item\_page*)* return html\_obj  
  
 def next\_page*(*self, chrome\_obj*)*:  
 *"""加载更多数据"""* print*(*"\033[1;36m>>正在爬取数据...\033[0m"*)* for j in range*(*self.Page*)*:  
 for i in range*(*4*)*: # 滑动延时 模仿读新闻  
 time.sleep*(*random.randint*(*0, 1*))* try: # 捕捉异常，排除因为没有加载进度条而报错的bug  
 chrome\_obj.execute\_script*(* f'document.documentElement.scrollTop=*{(*i + 1*)* \* 2000*}*'*)* except Exception as e:  
 continue  
 try:  
 click\_obj = chrome\_obj.find\_element*(*By.ID, "sceneGetMore"*)* # 定位 下页  
 click\_obj.click*()* # 点击  
 except Exception as e:  
 continue  
  
 return chrome\_obj  
  
 def Go*(*self*)*:  
 *"""开始爬取新闻"""* chrome\_obj = self.chrome\_obj*()* # 获取一个 Chrome 对象  
  
 chrome\_obj = self.next\_page*(*chrome\_obj*)* html\_obj = self.get\_page\_source*(*chrome\_obj*)* car\_, score\_, price\_, tags\_ = self.parse\_*(*html\_obj*)* for i in range*(*len*(*car\_*))*:  
 print*(*"\033[1;36m" + str*(*i + 1*)* + ". 车名:" + car\_*[*i*]* + " 评分:" + score\_*[*i*]* + " 价格:" + price\_*[*i*]* + " 特点:" +  
 tags\_*[*i*]* + "\033[0m"*)* self.save\_data*(*car\_, score\_, price\_, tags\_*)* def parse\_*(*self, html\_obj*)*:  
 *"""解析新闻列表页面"""* print*(*"\033[1;36m>>正在解析数据...\033[0m"*)* car\_ = html\_obj.xpath*(*'//li[@class="car-item"]/span[1]/text()'*)* # # 车名的获取  
  
 score\_ = html\_obj.xpath*(*'//div[@class="car-star"]/span[3]/text()'*)* # 评分的获取  
  
 price\_ = html\_obj.xpath*(*'//span[@class="car-price"]/text()'*)* # 价格的获取  
  
 tags\_1 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[1]/text()'*)* # 评论1的获取  
 tags\_2 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[2]/text()'*)* # 评论2的获取  
 tags\_3 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[3]/text()'*)* # 评论3的获取  
 tags\_4 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[4]/text()'*)* # 评论4的获取  
 tags\_5 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[5]/text()'*)* # 评论5的获取  
 tags\_6 = html\_obj.xpath*(*'//div[@class="car-tags"]/span[6]/text()'*)* # 评论6的获取  
 tags\_ = *[]* for i in range*(*len*(*tags\_1*))*:  
 tags\_.append*(*tags\_1*[*i*]* + "," + tags\_2*[*i*]* + "," + tags\_3*[*i*]* + ","  
 + tags\_4*[*i*]* + "," + tags\_5*[*i*]* + "," + tags\_6*[*i*])* # 合并6条评论  
  
 print*(*"\033[1;36m>>数据解析完毕...\033[0m"*)* return car\_, score\_, price\_, tags\_ # 返回Chrome对象和数据列表  
  
 def save\_data*(*self, car\_, score\_, price\_, tags\_*)*:  
 *"""存储数据到 Excel 表格"""* book = xlwt.Workbook*(*encoding='utf-8', style\_compression=0*)* sheet = book.add\_sheet*(*'SUV紧凑型车', cell\_overwrite\_ok=True*)* col = *(*"车名", "评分", "价格", "特点"*)* for j in range*(*len*(*col*))*:  
 sheet.write*(*0, j, col*[*j*])* # 构建表头  
  
 for i in range*(*1, len*(*car\_*)* + 1*)*: # 一列一列写入数据  
 sheet.write*(*i, 0, car\_*[*i - 1*])* # 车名  
 sheet.write*(*i, 1, score\_*[*i - 1*])* # 评分  
 sheet.write*(*i, 2, price\_*[*i - 1*])* # 价格  
 sheet.write*(*i, 3, tags\_*[*i - 1*])* # 特点  
  
 book.save*(*'SUV紧凑型车.xls'*)* print*(*"\033[1;36m>>数据下载完毕...\033[0m"*)*if \_\_name\_\_ == '\_\_main\_\_':  
 Spider = SUV*()* Spider.Go*()*

1. 运行效果

可以看到代码成功实现了获取口碑频道SUV紧凑型车全部车名，价格，特点，评分的数据，并保存到了Excel。



文本

描述已自动生成